



1994

## The Abstraction-Filtration Test: Determining Non-Literal Copyright Protection for Software

Stephen H. Eland

Follow this and additional works at: <https://digitalcommons.law.villanova.edu/vlr>



Part of the [Intellectual Property Law Commons](#)

---

### Recommended Citation

Stephen H. Eland, *The Abstraction-Filtration Test: Determining Non-Literal Copyright Protection for Software*, 39 Vill. L. Rev. 665 (1994).

Available at: <https://digitalcommons.law.villanova.edu/vlr/vol39/iss3/3>

This Note is brought to you for free and open access by Villanova University Charles Widger School of Law Digital Repository. It has been accepted for inclusion in Villanova Law Review by an authorized editor of Villanova University Charles Widger School of Law Digital Repository.

1994]

## Notes

### THE ABSTRACTION-FILTRATION TEST: DETERMINING NON-LITERAL COPYRIGHT PROTECTION FOR SOFTWARE

*Computer Associates International, Inc. v. Altai, Inc.* (1992)

#### I. INTRODUCTION

The United States is a world leader in computer technology.<sup>1</sup> In 1989, United States software suppliers accounted for more than sixty percent of the software sold throughout the world.<sup>2</sup> The success of industries served by software depends upon the quality and efficiency of the software available.<sup>3</sup> Therefore, to promote the success of these industries,

1. UNITED STATES OFFICE OF TECHNOLOGY ASSESSMENT, FINDING A BALANCE: COMPUTER SOFTWARE, INTELLECTUAL PROPERTY AND THE CHALLENGE OF TECHNOLOGICAL CHANGE 94 (1992) [hereinafter FINDING A BALANCE] ("By almost any measure, the United States has a premier role, both as producer and consumer of software.").

2. *Id.* In 1990, United States companies accounted for more than 70% of the sales in European markets for personal computer software. *Id.* at 96. The United States Department of Commerce estimated that global sales totalled over \$65 billion in 1989. *Id.* at 94. Although the market is dominated by several large software companies, revenues from United States independent software developers exceeded \$25 billion by 1988. *Id.* at 95. Of these revenues, 40% were from foreign sales. *Id.* The following is a list of the top 10 software companies in the North American Market:

Company	Estimated Revenues	
	1989	1990
	(\$000,000s)	
IBM	8,424	9,952
Microsoft	821	1,323
Computer Assoc.	1,290	1,311
Digital	825	810
Oracle	554	702
Lotus	516	635
Unisys	875	600
D&B Software	450	539
WordPerfect	281	452
Novell	288	388

*Id.* at 94.

3. ANTHONY L. CLAPES, SOFTWARE, COPYRIGHT, AND COMPETITION 19-20 (1989). Clapes summed up the computer industry's importance stating that: From today into the foreseeable future, it is obvious that the success of a country's computer industry on the world market will depend in large measure on the vitality of the software segment of that industry. Just as true, though, . . . is the fact that the success of other industries—particularly service industries such as banking and brokerage—will depend on

(665)

the United States must retain its prominent position in the global software market.<sup>4</sup>

Software piracy—the unauthorized duplication of software—threatens the United States' prominent position. Worldwide software piracy costs the United States software industry ten to twelve billion dollars annually in lost revenues.<sup>5</sup> In reaction to this pervasive problem, the software industry has sought the protection of patent, copyright and trade secret laws.<sup>6</sup> Of the three, copyright law is the most widely used form of legal protection for software.<sup>7</sup>

For protection to be valuable to software developers, the scope of copyright protection must be well-defined.<sup>8</sup> A well-defined scope of copyright protection increases the protection's value to developers in two ways. First, developers will be better able to estimate whether certain software will be economically feasible to develop. Because elements of a program that are not protected can be freely copied by others, these unprotected elements will be less profitable to produce. Second, software developers can reduce their development costs by copying unprotected

---

the quality and the efficiency of the software that controls their computerized facilities.

*Id.*

4. *Id.* ("All of the major industrialized nations have recognized that the business of writing software is critical to the competitiveness of their economies.").

5. FINDING A BALANCE, *supra* note 1, at 98. Unauthorized copying of business software for personal computers in the United States alone amounted to an estimated \$2.2 billion in lost revenues for software publishers. *Id.* at 100. It is estimated that for every legal software package in use in the United States, there is also one unauthorized copy in use. *Id.* In Europe, the estimates of software piracy are worse. *Id.* at 102. The following data shows that in major European countries, for every legitimate software sale one to four pirated copies are also sold:

Country	1989 Estimate (\$ 000,000s)	
	Software Sales	Value of Pirated Software
United Kingdom	795	439
France	605	628
W. Germany (formerly)	581	1440
Italy	190	768
Sweden	188	151

*Id.*

In Eastern Asia, estimates are even worse. *Id.* at 102-03. Estimates show that in 1990, 75% of the software used in South Korea was pirated. *Id.* at 102. Additionally, it has been estimated that 97% of the software in use in Thailand was pirated. *Id.* at 103.

6. *Id.* at 97. Government protection attempts to give software developers adequate market incentives to invest the time and resources required to produce innovative products. *Id.*

7. FINDING A BALANCE, *supra* note 1, at 56.

8. See D.C. Toedt III, *Why are We Re-inventing the Wheel? Arguments Against Copyright Protection for Command-Driven Software Interfaces*, 5 SOFTWARE L.J. 385, 388 (1992) (arguing that while patent law should be exclusive protection for software, fear, uncertainty and doubt about scope of copyright protection is significant cost to developers).

elements of other authors' programs<sup>9</sup>, without unnecessary and redundant development by a second author.<sup>10</sup> Without a clear definition of the scope of copyright protection, a developer may rewrite an element of another author's program rather than copying the element and risking copyright infringement.<sup>11</sup>

Under current copyright law, as embodied by the Copyright Act, the scope of protection for computer programs is poorly-defined.<sup>12</sup> Courts interpreting the Copyright Act have difficulty defining the scope of copyright protection for software because computer programs are utilitarian in nature, and computer technology advances at a rapid rate.<sup>13</sup> Computer programs are "literary works" within the meaning of the Copyright Act, and are, therefore, protected under the Act.<sup>14</sup> Literary works are

---

9. FINDING A BALANCE, *supra* note 1, at 153-54. There is a large amount of redundant work involved in writing common program elements or routines. *Id.* The concept of "reuse" has developed to avoid "re-inventing the wheel" when creating a new program. *Id.* at 153. It is estimated that of the 15.3 billion lines of computer code written in 1990, 60-70% involved "generic computer tasks like data entry, storage, and sorting." *Id.* at 154.

10. See, e.g., Toedt, *supra* note 8 (discussing waste and lack of competition in software industry due to uncertainty of copyright protection for software and concluding that patent law is more appropriate form of protection).

11. Toedt, *supra* note 8, at 388. Toedt stated that lawyers can not "predict with any confidence whether a would-be competitor can lawfully copy another software company's product." *Id.* Toedt also describes a situation involving a client who decided not to pursue a new software opportunity. *Id.* The client was unsure of the copyright implications, but could not afford the \$100,000 to \$500,000 in legal fees needed to defend the program from copyright infringement claims. *Id.*

12. RAYMOND T. NIMMER, THE LAW OF COMPUTER TECHNOLOGY ¶ 1.01, at 1-3 (2d ed. 1992) [hereinafter NIMMER ON COMPUTER LAW] (observing that decision to extend copyright protection to computer programs "established immediate legal protection . . . but yielded difficult and as yet unanswered questions about how copyright laws [will] mesh with product development and competition in a technology industry"); Toedt, *supra* note 8, at 385-86 (asserting that courts have forced copyright law into software context and have created fear, uncertainty and doubt in software industry).

13. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 704 (2d Cir. 1992). ("The essentially utilitarian nature of a computer program further complicates the task of distilling its idea from its expression.").

Computer technology has increased dramatically in efficiency while decreasing sharply in cost. CLAPES, *supra* note 3, at 24. For example, in 1952 it cost almost \$300 to perform one million processing operations. *Id.* Thirty years later, the same work cost only one thousandth of one cent. *Id.* Not only has the processing become cheaper, but the computation speed is one billion times faster. *Id.* Additionally, when Congress amended the Copyright Act in 1980, it could not foresee the personal computer revolution that put a powerful computer in almost every home and office. FINDING A BALANCE, *supra* note 1, at 22.

14. 17 U.S.C. § 102(b) (1988) (historical note) (noting that literary works include "computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves").

protected even if they are utilitarian.<sup>15</sup> However, a fundamental precept of copyright law is that while expressions are protectable, ideas are not. This distinction between idea and expression is complicated by software's inherent utility.<sup>16</sup> The difficulty in making the distinction between idea and expression is exacerbated by the rate at which "technical advance[s] tumble after one another in blinding succession."<sup>17</sup> Although copyright law has been altered through the years to adapt to changing technology,<sup>18</sup> the special challenges of computer technology may present problems that the Copyright Act is not capable of handling.<sup>19</sup> However,

---

15. FINAL REPORT OF THE NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS 21 (1979) [hereinafter FINAL REPORT] (stating that copyright protection is not denied simply because a work has utilitarian aspects). Utilitarian works are works that achieve a useful result. Steven R. Englund, Note, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 MICH. L. REV. 866, 893 (1990).

16. *Computer Assocs.*, 982 F.2d at 704. The Second Circuit noted that: "The variations of expression found in purely creative compositions . . . are not directed towards practical application. . . . Thus, compared to aesthetic works, computer programs hover even more closely to the elusive boundary line described in § 102(b)." *Id.* The major difficulty raised by the utilitarian nature of computer programs is the idea/expression distinction. This distinction arises when determining whether an allegedly infringing work is substantially similar to the protectable aspect of a copyrighted work. *Id.* The district court in *Computer Assocs.* noted that "[i]n the context of computer programs, many of the familiar tests for similarity prove to be inadequate, for they were developed historically in the context of artistic and literary, rather than utilitarian, works." *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 775 F. Supp. 544, 558 (E.D.N.Y. 1991), *aff'd*, 982 F.2d 693 (2d Cir. 1992). In addition, Nimmer stated that: "The difficulties in applying the traditional substantial similarity test to computer programs are exacerbated by the fact that computer programming is as much a science as an art." 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03[F], at 13-78.27 (1992) [hereinafter NIMMER ON COPYRIGHT].

17. CLAPES, *supra* note 3, at 25 (1989).

18. *Id.* at 16 ("American copyright law has consistently flexed and expanded to embrace new technologies in which works of authorship have been embodied.").

This flexibility in the face of changing technology can be seen by the numerous congressional changes to the Copyright Act. *Id.* at 15. In 1790, the Copyright Act only covered maps, charts and books. *Id.* In 1802, Congress amended the Act so that no one "shall invent and design, engrave, etch or work . . . prints" of another artist. *Id.* During the mid-1800s, Congress changed the Act three times, adding musical compositions, photographs, paintings, drawings and other designs of fine arts. *Id.* In 1909, the Act was revamped to account for new media and expression, and shortly thereafter, motion pictures were added. *Id.* To protect the audio industry from the ease of copying afforded by audio tape and tape recorders, Congress added sound recordings to the Act's protection. *Id.* Finally, in 1976, computer programs were added to the Copyright Act. *Id.* This change was made more explicit in the 1980 amendment that incorporated the recommendations of the National Commission on New Technological Uses of Copyrighted Works. *Id.*

19. *Computer Assocs.*, 982 F.2d at 712 (observing that because computer programs are both literary expressions and functional utilitarian works, Copyright Act may afford weak protection).

until Congress modifies existing law or creates a sui generis law,<sup>20</sup> courts

20. Sui generis is a latin term meaning "of its own kind or class." BLACK'S LAW DICTIONARY 1434 (6th ed. 1990). In the present context, it means a new law that is not part of the patent or copyright laws. FINDING A BALANCE, *supra* note 1, at 7-8.

The notion of a new form of protection specifically designed to address computer software was advanced by the Second Circuit in *Computer Associates*: "Generally, we think that copyright registration—with its indiscriminating availability—is not ideally suited to deal with the highly dynamic technology of computer science. Thus far, many of the decisions in this area reflect the courts' attempt to fit the proverbial square peg in a round hole." *Computer Assocs.*, 982 F.2d at 712.

One of the major difficulties of present copyright law is that competitors are unsure of the scope of the law's protection. "Firms in the software market are probably less than pleased at the prospect of being the guinea pigs in this common law process and want to know the answers to protection questions now." Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1121, 1153 (1990). A sui generis law may give increased certainty in the field about what is protected, and under what conditions protection is available. *Id.* at 1152.

One commentator has noted that software is an example of a "legal hybrid" that falls between copyright and patent law. FINDING A BALANCE, *supra* note 1, at 27. Yet, legal protection is necessary in order to encourage innovation. *Id.* Unfortunately, patent laws have rigorous requirements that only protect a small portion of such works, while copyright law, with its lengthy period of protection, reflects cultural policies that are largely irrelevant to the needs of a competitive market. *Id.* The United States Office of Technology Assessment's report noted that the current developments in the copyright law may be adequate, however, "[d]espite the advantages of incremental accommodation within the copyright law, there may be a point where it becomes preferable to augment or complement the existing framework rather than extend the scope of copyright to fit software." *Id.* at 29. Additionally, because protection can be tailored to meet the computer industry's needs, a sui generis law would protect software without distorting patent or copyright laws. Samuelson, *supra*, at 1152.

One commentator has proposed a system of protection for noncode aspects of software which has five principal features:

- (1) registration without a determination as to the technical merits or validity of the work unless and until it is challenged in an infringement action,
- (2) conditioning protection on a level of creativity or technical advance between the patent system's high standard of unobviousness and the de minimis originality standard for copyright protection,
- (3) a period of protection between five and twenty years,
- (4) a set of exclusive rights that include the right to control the production, sale and use of the work, tempered by the exhaustion doctrine and fair use defense, and
- (5) a test of infringement requiring that a person of ordinary skill in the computer field would recognize that the allegedly infringing work is functionally interchangeable with protected subject matter, or that it could be devised from protected subject matter with routine effort and insubstantial expense.

Richard H. Stern, *The Bundle of Rights Suited to New Technology*, 47 U. PITT. L. REV. 1229.

There are three problems with a sui generis law: (1) determining what the law should cover; (2) uncertainty that will arise during the transition period while caselaw develops and (3) lack of established international treaty structure. FINDING A BALANCE, *supra* note 1, at 31. Additionally, any law that is created today may be outdated tomorrow, because Congress can not foresee a new technology.

The first problem mentioned above is similar to the major source of strife facing courts under the current law. In determining the balance between the

must utilize existing copyright doctrines to determine the scope of protection for computer programs.<sup>21</sup>

Courts have addressed two aspects of copyright protection for software: literal and non-literal. While many courts have discussed and defined literal protection, until recently, very few courts have discussed non-literal protection for software. As the value of software's non-literal elements has increased over the years, developers have become more concerned about the scope of copyright protection for non-literal elements. One court that attempted to clarify the scope of copyright protection for software's non-literal aspects was the United States Court of Appeals for the Second Circuit in *Computer Associates International, Inc. v. Altai, Inc.*<sup>22</sup> Prior to *Computer Associates*, the United States Court of Appeals for the Third Circuit in *Whelan Associates v. Jaslow Dental Laboratory*<sup>23</sup> was the only other court that had dealt squarely with the issue of non-literal protection for computer programs.

In *Computer Associates*, Altai, a software developer, hired a programmer from its competitor, Computer Associates (CA).<sup>24</sup> When rewriting one of Altai's programs, the programmer copied large portions of a CA program.<sup>25</sup> After CA informed Altai of the copying, Altai rewrote the program.<sup>26</sup> Even though the rewritten program eliminated all of the literal aspects of CA's program, CA insisted that Altai's program infringed upon CA's program because the non-literal elements of CA's program remained in Altai's program.<sup>27</sup> The United States District Court for the Eastern District of New York upheld CA's claim of literal infringement as to the copies that Altai had sold containing the literal elements of CA's

---

existing policies of protection and monopolization, Congress should consider several significant changes that have occurred since the 1980 amendment including:

[T]he "PC Revolution" and explosive growth in markets for personal computers and packaged software; widespread use of computers and software by nonprogrammers and the corresponding market importance of user interfaces; increased barriers to entry by small firms and a trend toward centralized software-publishing houses that acquire rights to software and then distribute and market it, paying royalties to the program developers; maturity of the software industry and increasing firm size (through growth, acquisition, and consolidation); and increasing industry concentration, especially when considering submarkets like PC applications.

*Id.* at 22.

21. *Computer Assocs.*, 982 F.2d at 712. Even though software may strain the copyright laws, it is important that courts do not "impair the overall integrity of copyright law." *Id.*

22. 982 F.2d 693, 712 (2d Cir. 1992).

23. 797 F.2d 1222 (3d Cir. 1986).

24. *Computer Assocs.*, 982 F.2d at 699-700.

25. *Id.* at 700.

26. *Id.*

27. *Id.*

program, but the district court denied CA's claim of non-literal infringement.<sup>28</sup>

On appeal, the Second Circuit addressed the issue of non-literal infringement and attempted to define the scope of copyright protection for non-literal aspects of computer programs. The Second Circuit rejected the Third Circuit's approach to non-literal protection that was developed in *Whelan*.<sup>29</sup> Instead, the Second Circuit created the Abstraction-Filtration test.<sup>30</sup> The Abstraction-Filtration test is a method for segregating the protectable elements of a computer program and determining whether the protected elements have been infringed upon by another work. The test is based on established principles of copyright law and provides courts with significant discretion when applying it.<sup>31</sup>

This Note concludes that the Abstraction-Filtration test is inherently flexible, thus, enabling the test to adapt to various technological situations. This flexibility allows courts to decide cases equitably, but does not provide a firm rule upon which software developers can rely.

Part II of this Note discusses the technological, economic and legal framework of copyright law, which provide a basis for analyzing the Second Circuit's decision in *Computer Associates*. Part III of this Note presents the facts of *Computer Associates* and analyzes the court's reasoning in the case. Part IV then provides an analysis of the Abstraction-Filtration test and compares the Abstraction-Filtration test with the test developed by the Third Circuit in *Whelan*. Finally, Part V of this Note examines several cases decided subsequent to *Computer Associates* in which courts have adopted the Abstraction-Filtration test.

---

28. *Id.* at 701.

29. *Whelan Assocs. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir. 1986). In *Computer Associates Int'l, Inc. v. Altai, Inc.*, the district court rejected the *Whelan* test because it was based upon an outdated appreciation of computer science. 775 F. Supp. 544, 559-60 (E.D.N.Y. 1991). The Second Circuit affirmed the district court stating that the *Whelan* test "relies too heavily on metaphysical distinctions and does not place enough emphasis on practical considerations." *Computer Assocs.*, 982 F.2d at 706.

30. *Computer Assocs.*, 982 F.2d at 706-12. The test examines an original work and distinguishes the work's ideas from its expression. *Id.* at 706-07. The test then filters out unprotectable expression. Finally, the protectable expression is compared with the allegedly infringing work. *Id.* at 707-12.

31. *Id.* at 706. The Abstraction-Filtration test does not break any new ground. *Id.* Rather, the test utilizes established copyright doctrines such as merger, *scenes à faire* and public domain. *Id.*

The Second Circuit recognized that the test must provide district courts with flexibility because judicial decision-making often is unable to keep pace with the rapid rate at which computer technology advances. *Id.* The Second Circuit introduced its opinion by noting that "where the technology in question does not allow for a literal application of the procedure we outline below, our opinion should not be read to foreclose the district courts of our circuit from utilizing a modified version." *Id.*



## II. BACKGROUND

To protect software developers adequately, courts must not only comprehend developers' needs, but also must understand the technology underlying the software.<sup>32</sup> Moreover, courts must recognize the numerous economic considerations that affect the scope of copyright protection for software.<sup>33</sup> In developing these critical issues, Part II-A of this Note introduces the basic precepts of copyright law that must be applied to an economic and technological framework. Part II-B then explains the process involved in developing software and Part II-C introduces the economic considerations that affect the amount of protection required. Finally, Part II-D of this Note discusses the underlying case law regarding the protection of non-literal aspects of computer programs.

## A. General Copyright Law Framework

A prima facie copyright infringement case requires proof of two elements: (1) ownership of a valid copyright and (2) copying a copyrighted work.<sup>34</sup> Because direct evidence of copying is rarely available, copying can be proved inferentially by demonstrating that the defendant had access to the allegedly infringed work and that the defendant's work is substantially similar to the plaintiff's work.<sup>35</sup>

Copyright infringement may occur even if a literal copy of the original work is not made.<sup>36</sup> Yet, by limiting the level of copyright protection to literal copies, a plagiarist may avoid the copyright by making only slight changes to the original.<sup>37</sup> However, because copyright law does not pro-

---

32. NIMMER ON COMPUTER LAW, *supra* note 12, ¶ 1.03[1], at 1-10 ("Application of even basic copyright concepts to computer software requires understanding of the multiple functions that software performs and the multiple forms in which it is found.")

33. Peter S. Mennell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1058-70 (1989) (developing economic framework for analyzing copyright protection for computer programs).

34. *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 775 F. Supp. 544, 555 (E.D.N.Y. 1991), *aff'd*, 982 F.2d 693 (2d Cir. 1992). Registering a work with the Register of Copyrights "before or within five years after first publication of the work . . . constitute[s] prima facie evidence of the validity of the copyright." 17 U.S.C. § 412(c) (1988).

35. *Whelan Assocs. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1231-32 (3d Cir. 1986) (citing *Roth Greeting Cards v. United Card Co.*, 429 F.2d 1106, 1110 (9th Cir. 1970) and *Ferguson v. National Broadcasting Co.*, 584 F.2d 111, 113 (5th Cir. 1978)); *see also* *Feist Pub., Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991) (concluding that plaintiff failed to show substantial similarity between defendant's work and protected aspects of plaintiff's work); *Walker v. Time Life Films, Inc.*, 784 F.2d 44, 48 (2d Cir.), *cert. denied*, 476 U.S. 1159 (1986).

36. *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 701 (2d Cir. 1992) ("As a general matter, and to varying degrees, copyright protection extends beyond a literary work's strictly textual form to its non-literal components.")

37. *Nichols v. Universal Pictures Co.*, 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931). In *Nichols v. Universal Pictures Co.*, Justice Learned Hand stated that it is "essential to any protection of literary property . . . that the

tect ideas, difficulties arise in determining which non-literal components of a work are protectable by copyright.<sup>38</sup> Although the stealing of ideas contradicts notions of justice, courts have recognized that lack of protection for ideas is a fundamental principal of copyright law.<sup>39</sup>

The denial of protection for ideas, known as the idea/expression distinction, is the primary limitation on the scope of copyright protection.<sup>40</sup>

---

right can not be limited literally to the text, else a plagiarist would escape by immaterial variations." *Id.*

Frequently the copyright of an original work is infringed even though there was no substantial similarity between the literal elements of the original work and the copy. *Whelan*, 797 F.2d at 1234. A copyright may be infringed by merely copying its plot or plot devices. *Id.*; see, e.g., *Computer Assocs.*, 982 F.2d at 701 (citing *Horgan v. Macmillan*, 789 F.2d 157, 162 (2d Cir. 1986)) (noting that "to varying degrees, copyright protection extends beyond a literary work's strictly textual form to its non-literal components"); *Twentieth Century-Fox Film Corp. v. MCA, Inc.*, 715 F.2d 1327, 1329 (9th Cir. 1983) (noting that numerous distinctive plot similarities between *Battlestar Galactica* and *Star Wars* were possible bases for finding copyright infringement); *Sid & Marty Kroft Television Prod. v. McDonald's Corp.* 562 F.2d 1157, 1167 (9th Cir. 1977) (stating that similarities between McDonaldland characters and H.R. Pufnstuf characters may be established by "total concept and feel").

38. *Mazur v. Stein*, 347 U.S. 201, 217 (1954). The Supreme Court stated: "Unlike a patent, a copyright gives no exclusive right to the art disclosed; protection is given only to the expression of the idea—not the idea itself." *Id.* This precept has been mandated by Congress in the Copyright Act which states: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. § 102(b) (1988). Additionally, Congress has stated that a "[c]opyright does not preclude others from using ideas or information revealed by the author's work." H.R. REP. NO. 1476, 94th Cong., 2d Sess. 54 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5670.

A book describing the rules of a game is an example of the difference between protectable expressions and unprotectable ideas. "[O]ne may not adopt and republish or redistribute copyrighted game rules, but the owner has no power to prevent others from playing the game." FINAL REPORT, *supra* note 15, at 20 (footnote omitted).

39. CLAPES, *supra* note 3, at 3 ("The principle that one's precious and hard-won ideas, once published, are not protected from copying collides with the creative but orderly mind's sense of justice, and is rejected time and time again."); *Computer Assocs.*, 982 F.2d at 703 (noting that "a fundamental principle of copyright law is that a copyright does not protect an idea"); see also *Cooling Sys. & Flexibles, Inc. v. Stuart Radiator, Inc.*, 777 F.2d 485, 491 (9th Cir. 1985) ("Copyright law never protects the facts and ideas contained in published works."). Intrinsically utilitarian functions can be denied copyright protection "except to the extent that its artistic features can be identified separately and are capable of existing independently as a work of art." *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1007, 1023 (N.D. Cal. 1992) (quoting *Fabrica Inc. v. El Dorado Corp.*, 697 F.2d 890, 893 (9th Cir. 1983)).

40. *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465, 1472 (9th Cir.), *cert. denied*, 113 S. Ct. 198 (1992). In *Brown Bag Software v. Symantec Corp.*, the Ninth Circuit stated that the program's functions "constitute the idea of the outlining program [and the] expression of the ideas inherent in the features are . . . distinct." *Id.* In the report by the National Commission on New Technological Uses of Copyrighted Workers (CONTU), the commission stated that the idea/

Such a limitation on idea protection is often based on fine line distinctions that must be made on a case-by-case basis.<sup>41</sup> The limitation on idea protection is especially important in the computer science field because growth and advancement depends upon free access to existing ideas.<sup>42</sup> Therefore, for proper growth and advancement to occur, copyright protection should not extend to information that is already in the public domain.<sup>43</sup>

Separating ideas from expression is a difficult, fact-sensitive determination. Judge Learned Hand described the distinction between idea and expression as an "abstract continuum," which later became known as abstraction.<sup>44</sup> There is a point along this continuum where a work is no

---

expression distinction should be used to determine what aspects of a computer program are copyrightable. FINAL REPORT, *supra* note 15, at 44. The report stated that "[s]ection 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyright element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of copyright law." *Id.* at 19 (footnote omitted). The Office of Technology Assessment noted that this "traditionally 'fuzzy' line" is confounded in computer software cases. *Id.* at 22-23.

41. *Nichols*, 45 F.2d at 121. In *Nichols*, Judge Hand observed that "[n]obody has been able to fix that boundary, and nobody ever can." *Id.* In a later case, he stated that "[o]bviously, no principle can be stated as to when an imitator has gone beyond copying the 'idea' and has borrowed its 'expression'." *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir. 1960). Ordinarily, the term "expression" connotes one human communicating with another. NIMMER ON COMPUTER LAW, *supra* note 12, ¶ 1.02, at 1-5. In the copyright context of idea/expression, the term has a much more technical definition. *Id.* "Expression exists if an author or technician makes personal choices with a goal in mind and records them on a medium of expression . . . ." *Id.* One author has suggested that the defense of copying ideas rather than expression "throws the court into a gray area and [allows the defendant to] put forward arguments not really directed to the idea-expression dichotomy at all, but to other reasons why the defendant should be let off the hook." CLAPES, *supra* note 3, at 66. This idea is consistent with Judge Hand's statement that decisions must inevitably be ad hoc. *Peter Pan Fabrics*, 274 F.2d at 489.

42. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.30.

43. *Id.* at 13-80 ("An enormous amount of public domain software exists in the computer industry, perhaps to a much greater extent than is true of other fields.").

44. *Nichols*, 45 F.2d at 121. Judge Hand noted in *Nichols*:

Upon any work . . . a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the work is about, and at times might consist of only its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the [author] could prevent the use of his "ideas," to which, apart from their expression, his property is never extended.

*Id.*

In *Lotus Development Corp. v. Borland International, Inc.*, the United States District Court for the District of Massachusetts gave an example of abstraction:

- (1) Lotus 1-2-3 is an electronic spreadsheet.
- (2) It is a menu-driven electronic spreadsheet.

longer protected because it is more an idea than an expression.<sup>45</sup> Abstraction is not a test; it is merely a manner of describing the difficulties that courts face in trying to avoid either extreme of the continuum.<sup>46</sup> Because abstraction determines the scope of a plaintiff's copyright protection, the line along the continuum that separates the work's idea from its expression must be drawn by balancing the competing policies of copyright protection.<sup>47</sup>

In addition to the idea/expression distinction, the scope of copyright protection is limited by the merger doctrine.<sup>48</sup> The merger doctrine broadens the idea/expression dichotomy by excluding copyright protec-

(3) Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command.

(4) Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the specific spreadsheet operations available in Lotus 1-2-3 are accessible through the paths of the menu command hierarchy.

(5) Finally, one may conceive of the interface as that precise set of menu commands selected by Lotus, arranged hierarchically precisely as they appear in 1-2-3.

799 F. Supp. 203, 216 (D. Mass. 1992). The district court began in step one with the most general description of the program. This is undoubtedly an idea. The court then progressed toward the actual expression of the program. Somewhere along this line, the unprotectable idea becomes protectable expression.

45. *Nichols*, 45 F.2d at 121. "Judge Hand's abstraction test implicitly recognizes that any given work may consist of numerous ideas and expressions, ranging from the most general statement of 'what the [work] is about,' to specific choices of words or code." *NIMMER ON COPYRIGHT*, *supra* note 16, at § 13.03[F], at 13-78.34 (quoting *Nichols v. Universal Pictures Co.*, 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931)).

46. *Nash v. CBS, Inc.*, 899 F.2d 1537, 1540 (7th Cir. 1990).

47. *NIMMER ON COPYRIGHT*, *supra* note 16, § 13.03[F], at 13-78.34. The line drawn between levels of abstraction is "a pragmatic one, drawn not on the basis of some metaphysical property of 'ideas,' but by balancing the need to protect the labors of authors with the desire to assure free access to ideas." *Id.*

48. In 1879, the Supreme Court established the doctrine of merger in *Baker v. Selden*, 101 U.S. 99 (1879). Plaintiff Selden had copyrighted a book that improved upon an established system of bookkeeping. *Id.* at 100. The book contained several illustrations and diagrams describing the system. *Id.* Defendant Baker wrote a book regarding substantially the same system. *Id.* at 101. Selden claimed that "the ruled lines and headings, given to illustrate the system, are part of the book, and, as such, are secured by the copyright; and that no one can make or use . . . the same system, without violating the copyright." *Id.* The court reasoned that only a patent could protect the system, and analogized the situation to a book describing medicine. *Id.* at 102. If a doctor desires to gain exclusive right to the manufacture and sale of a medicine, the doctor must obtain a patent for the compound or process. *Id.* at 102-03. "He may copyright [a] book if he pleases; but that only secures him the exclusive right of printing and publishing his book." *Id.* at 103.

tion for those aspects of a work that are necessarily incident to the work's underlying concept.<sup>49</sup> Thus, when there is essentially only one way to express an idea, the idea and its expression are inseparable and copyright is no bar to copying that expression.<sup>50</sup> This is true for both literal and non-literal similarity.<sup>51</sup> In some situations, the merger doctrine may apply even though there are several ways to express an idea.<sup>52</sup>

Application of the merger doctrine depends on the scope of the idea.<sup>53</sup> Generally speaking, as the number of expressions for an idea de-

In a similar situation, the rules of a sweepstakes were not protectable by copyright, because the expression was inseparable from the idea of the contest. *Morrissey v. Proctor & Gamble, Co.*, 379 F.2d 675, 678-79 (1st Cir. 1967).

49. *Baker*, 101 U.S. at 104. The case does not impose a limit on the ability to copyright works that express ideas, systems, or processes. Rather, "[t]he case properly stands for the proposition that using the system does not infringe the copyright in the description." FINAL REPORT, *supra* note 15, at 19.

50. *Mason v. Montgomery Data, Inc.*, 967 F.2d 135, 138 (5th Cir. 1992) ("[W]hen there is essentially only one way to express an idea, 'copying the expression will not be barred, since protecting the expression in such circumstances would confer a monopoly of the idea upon the copyright owner free of the conditions and limitations imposed by the patent law.'" (quoting *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971))); *Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 606 (1st Cir. 1988) (noting that even when idea and expression are separable, scope of copyright protection may be restricted if modes of expressing idea are limited).

For instance, during a competition for government approval, a company added lines and mile markings to a United States Geographical Survey topographical map. *Kern River Gas Transmission Co. v. Coastal Corp.*, 899 F.2d 1458 (5th Cir.) *cert. denied*, 498 U.S. 952 (1990). A competitor copied the map with these lines and markings. *Id.* at 1464. During a copyright infringement suit, the plaintiff's "principal planning engineer testified that he could think of no other way to portray the idea of the pipeline's proposed location." *Id.* Relying upon this statement as reflected in the record of the district court, the Fifth Circuit reasoned that "[t]o extend copyright protection to the road maps would grant [plaintiff] a monopoly over the only approved pipeline route." *Id.* at 1464-65.

In some instances, the copyright owner may desire to establish that alternate means of expression exist in order to prove that the idea and expression have not merged. *See Pearl Sys., Inc. v. Competition Elecs., Inc.*, 1988 Copy. L. Dec. (CCH) ¶ 26,338 (S.D. Fla. 1988) (hiring party to produce same program with different expression). Conversely, a defendant may attempt to establish that such an idea is constrained to the particular expression. *NEC Corp. v. Intel Corp.*, 1989 Copy. L. Dec. (CCH) ¶ 26,379 (N.D. Cal. 1989) (setting up "clean room" and hiring a third party to demonstrate that even without access, program produced would be substantially similar to allegedly infringed program).

51. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-67 (citing *Landsberg v. Scrabble Crossword Game Players, Inc.*, 736 F.2d 485 (9th Cir. 1984) and *Atari, Inc. v. North American Philips Consumer Elec. Corp.*, 672 F.2d 607 (7th Cir. 1982)).

52. *Morrissey v. Proctor & Gamble Co.*, 379 F.2d 675, 679 (1st Cir. 1967). In these situations, if "the subject matter would be appropriated by permitting the copyrighting of its expression [then protection must be denied; otherwise the court would be forced to] recognize copyright as a game of chess in which the public can be checkmated." *Id.*

53. *Mason*, 967 F.2d at 140. Nimmer states that "as with the idea/expression dichotomy that pervades this area, application of the merger doctrine depends on

creases, the similarity necessary for infringement increases.<sup>54</sup> With regard to defining the underlying idea, a court should keep in mind the essential balance in copyright laws: a balance between free market competition and governmental protection.<sup>55</sup> To achieve this balance an idea should be defined broadly in situations involving factual or utilitarian works, and more narrowly in situations involving subjective matters of opinion.<sup>56</sup>

Copyright protection is further limited by the doctrine of *scènes à faire*.<sup>57</sup> *Scènes à faire* are incidents, characteristics or settings that are, as a practical matter, indispensable in the treatment of a given topic.<sup>58</sup> *Scènes à faire* are not protected by copyright because such protection would allow an author to monopolize a commonplace idea.<sup>59</sup> In a similar vein, copy-

---

... [how] one defines the 'idea' that merges with the subject expression." NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[B], at 13-78 (citing *Kregos v. Associated Press*, 937 F.2d 700, 707 (2d Cir. 1991)).

54. *Cooling Sys. & Flexibles, Inc. v. Stuart Radiator, Inc.*, 777 F.2d 485, 491 (9th Cir. 1985) (observing that "the fewer the methods for expressing an idea, the more the allegedly infringing work must resemble the copyrighted work in order to establish substantial similarity"). Additionally, the Ninth Circuit noted that when an idea and its expression merge, the work is only protected against identical copying. *Sid & Marty Kroft Television Prods. v. McDonald's Corp.*, 562 F.2d 1157 (9th Cir. 1977).

55. *Mason*, 967 F.2d at 140. The Fifth Circuit went on to say that "[i]n copyright law, an 'idea' is not an epistemological concept, but a legal conclusion prompted by notions—often unarticulated and unproven—of appropriate competition." *Id.* at 140 n.8 (quoting Jane C. Ginsburg, *No "Sweat"? Copyright and Other Protection of Works of Information after Feist v. Rural Telephone*, 92 COLUM. L. REV. 338, 346 (1992)). Following a similar reasoning, Nimmer observed that when an idea is "inseparably tied to a particular expression . . . , rigorously protecting the expression would confer a monopoly over the idea itself." NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[B], at 13-65.

56. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[B], at 13-68 (citing *Kregos v. Associated Press*, 937 F.2d 700, 707 (2d Cir. 1991)).

57. *See Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 606 (1st Cir. 1988) (denying copyright protection when idea and expression merge); *Whelan Assocs. v. Jaslow Dental Lab.*, 797 F.2d 1222, 1236 (3d Cir. 1986) ("It is well-settled doctrine that *scènes à faire* are afforded no copyright protection.").

58. *Whelan*, 797 F.2d at 1236 (quoting *Atari, Inc. v. North American Philips Consumer Electronics Corp.*, 672 F.2d 607, 616 (7th Cir.), *cert. denied*, 459 U.S. 880 (1982)). The phrase "*scènes à faire*" is a French dramatic literary phrase meaning "a scene which must be done." *Schwarz v. Universal Pictures Co.*, 85 F. Supp. 270, 275 (S.D. Cal. 1945). Therefore, because "it is virtually impossible to write about a particular historical era or fictional theme without employing certain 'stock' or standard literary devices, [the Second Circuit has] held that *scènes à faire* are not copyrightable as a matter of law." *Hoehling v. Universal City Studios, Inc.*, 618 F.2d 972, 979 (2d Cir.), *cert. denied*, 449 U.S. 841 (1980).

59. *Whelan*, 797 F.2d at 1236. "*Scènes à faire* are afforded no protection because the subject matter represented can be expressed in no other way than through the particular *scene à faire*. Therefore, granting a copyright 'would give the first author a monopoly on the commonplace ideas behind the *scènes à faire*.'" *Id.* (quoting *Landsberg v. Scrabble Crossword Game Players, Inc.*, 736 F.2d 485, 489 (9th Cir. 1984)).

right protection also does not extend to information that is already in the public domain,<sup>60</sup> even if it is incorporated into a copyrightable work.<sup>61</sup>

### B. Computer Programming

Three factors regarding computer programming affect the scope of protection needed for software's non-literal aspects: (1) the method of program development; (2) constraints on a program's structure and (3) the incremental progression of software technology. The first two factors, programming methodology and structure constraints, exemplify why the non-literal aspects of a program can be the most valuable aspects of a program. The third factor, software's incremental progression, provides an example of why increased non-literal protection may inhibit software "structure" and development.

Software development methodology is the first factor affecting the scope of protection for software's non-literal elements. Computer program development takes place in several steps.<sup>62</sup> Ordinarily, a programmer utilizes the "top-down" approach, whereby a general idea is broken down into numerous less complex ideas and then organized.<sup>63</sup> The programmer implements the "top-down" approach by:<sup>64</sup> (1) identifying the ultimate function; (2) decomposing the function into subroutines; (3) decomposing the subroutines further if needed; (4) arranging the

60. *Cooling Sys.*, 777 F.2d at 491 ("An author can claim to 'own' only an original manner of expressing ideas or an original arrangement of facts.").

61. *Sheldon v. Metro-Goldwyn Pictures Corp.*, 81 F.2d 49, 54 (2d Cir. 1936), *aff'd*, 309 U.S. 390 (1940); *see also* *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 710 (2d Cir. 1992) (holding that material in public domain is "free for the taking," and cannot be monopolized by single author even if material is incorporated into copyrighted work).

62. Congress has defined a computer program as: "[A] set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101 (1988).

63. *Whelan*, 797 F.2d at 1230; NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.31; Mennell, *supra* note 33, at 1055.

64. Mennell, *supra* note 33, at 1051. Variations of these steps, embodying the same ideas are found in the Second Circuit's opinion in *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 697-98 (2d Cir. 1992). *See* NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.32 (1992).

After a program is created, debugging and documentation begin. *Whelan*, 797 F.2d at 1231 n.21. Debugging is the process of removing errors. *Id.* For any program of at least moderate length, the process can be very time consuming, because syntax errors, as well as logic errors, must be corrected. *Id.* Syntax errors are mistakes in the actual form of a program statement. *Id.* For an explanation of the syntax that constrain computer programs, *see infra* note 66. After debugging, the software must be documented. *Whelan*, 797 F.2d at 1231 n.21. Documentation is material that a programmer provides for the user to explain how the program runs. The documentation must anticipate potential questions or problems that a user may encounter. *Id.*

subroutines into organizational or flow charts<sup>65</sup> and (5) writing the actual code.<sup>66</sup>

A program's literal aspects include only its actual code. On the other hand, a program's non-literal aspects comprise its structure, and include components such as general flowcharts, organization of the relationship between subroutines, parameter lists and macros.<sup>67</sup> In terms of time and effort, therefore, creating a program's structure is often the most demanding programming stage.<sup>68</sup>

---

65. During this stage, the problem is broken into subroutines and organized to show the relationship between these subroutines. Mennell, *supra* note 33, at 1055. Additionally, at this point, pseudo-code may be written to represent the subroutines' functions. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.32. Pseudo-code is:

[A] representation of program structure consisting of written statements which resemble actual source code. Pseudo-code, however, omits much of the detail of actual code, and is not restricted to any particular computer language. Thus, for example, a programmer writing pseudo-code might label a function "OUTPUT" which, when actually coded, would require commands to open files and print.

*Id.*

66. During this stage the subroutines are written in a "high-level" language, which is "a symbolic language, often using English words and common mathematical symbols, that humans can read." NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-117 n.271. This version of the program is typically denoted source code. *Id.* Although current high-level languages are easier to use than older "low-level" programming languages, the actual syntax of the computer instructions is very formal. CLAPES, *supra* note 3, at 62. Each computer language requires a certain form in order to execute a certain command. *Id.* Any deviation from this form will result in the computer being unable to understand the command. *Id.* These syntax errors can be as slight as using a comma in a statement rather than a semicolon. Source code is constrained to a strict syntax because a translating program must convert the source code into object code. *Id.* Source code is then converted into object code by the use of a compiler. *Id.* Object code is "a program expressed as binary numbers comprehensible to the computer, a pattern of ones and zeros that cause the computer to execute a coherent set of operations leading to a useful result." *Id.* at 31. It is almost impossible to gain a working understanding of a computer program, by analyzing its object code. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.26 n.271. Therefore, to protect its secrecy, most commercial software is only sold in its object code form. *Id.*

67. *Computer Assocs.*, 982 F.2d at 702. A macro is a "single instruction that initiates a sequence of operations or module interactions within the program." *Id.* at 698. For proper interaction between the main program and the subroutines, "interacting modules must share similar parameter lists so that they are capable of exchanging information." *Id.* A parameter list "refers to the form in which information is passed between modules (e.g. for accounts receivable, the designated time frame and particular customer identifying number) and the information's actual content (e.g. 8/91-7/92; customer No.3)." *Id.* at 697-98.

68. *Whelan, Inc.*, 797 F.2d at 1231 ("By far the larger portion of the expense and difficulty in creating computer programs is attributable to the development of the structure and logic of the program, and to debugging, documentation and maintenance, rather than to the coding."); NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.31 n.288 ("A significant part of the effort involved in virtually



If software's non-literal aspects are not protected, then developers will have less incentive to expend the time and effort needed to develop new software, because a competitor would be free to copy the original author's non-literal aspects. The competitor could then add the literal aspects and enter the market at a lower cost than the original author.

The methodology involved in developing software is not the only reason non-literal elements are valuable to programmers; constraints on the non-literal aspects increase a program's value and consequently affect the scope of necessary protection. Constraints include concerns regarding speed, efficiency and ease of use. When creating the structure, the programmer attempts to maximize the program's speed efficiency and ease of use.<sup>69</sup> Integrating these constraints into a structure may require considerable time and effort, which increases the value of the non-literal elements to the author. Therefore, similar to the first factor, a developer may not have sufficient incentive to develop non-literal elements if such elements are not protected.

Software development may also be inhibited if an author can, through legal protection, monopolize the only option that is workable because of speed, efficiency and ease of use constraints.<sup>70</sup> The concern about allowing an author to monopolize a program's only workable option is closely related to the third factor, determining the incremental progression of computer technology.<sup>71</sup> Because computer technology is

---

any programming project is spent in specifying adequate data structures and the means for manipulating them.").

69. *Computer Assocs.*, 982 F.2d at 698.

70. See NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.35. Nimmer states that "[a]lthough theoretically many ways may exist to implement a particular idea, efficiency concerns can make one or two choices so compelling as to virtually eliminate any other form of expression." *Id.* As an example, Nimmer notes the differences between two sorting techniques: bubble sort and quick sort. *Id.* at 13-78.36 n.306. For sorting a number of names(N) using bubble sort, the number of comparisons necessary is a function of the number of names squared ( $N^2$ ); using quick sort the number of comparisons necessary is a function of the product of N, and the natural log (e-based logarithm) of N ( $N\ln(N)$ ). *Id.* This means that for 1000 names, the number of comparisons necessary using bubble sort will be on the order of 1,000,000. *Id.* Using quick sort, the number of comparisons will be in the range of 7,000. *Id.* If an author is able to monopolize the use of the quick sort, subsequent authors who need to use a sorting method will be forced to use an inferior method.

71. Menell, *supra* note 33, at 1060. Computer technology advances by sequential improvements. *Id.* This method of advancement is different from that of other arts, and therefore, excessive protection could delay the design improvements, refinements and adaptation of the work to different uses. *Id.* The Third Circuit in *Whelan* disagreed, stating that this incremental development is not different from progress in other arts or sciences. *Whelan*, 797 F.2d at 1238. In light of such a rationale, the Third Circuit's approach should not be any different, because copyright has always attempted to recognize that advances in a field require authors to build upon the work of their predecessors. *Id.* Interestingly, the court went on to note that "[l]ong before the first computer, Sir Isaac Newton humbly explained that 'if [he] had seen farther than other men, it was because [he] stood on the shoulders of giants.'" *Id.* at 1238 n.33.

advanced by means of "stepping stones," innovation may require plagiarizing portions of a copyrighted work.<sup>72</sup> If authors can monopolize significant non-literal portions of their programs, subsequent authors will not be able to advance computer technology by building upon the work of prior authors.

### C. *Economic Background*

The Congressional grant of limited monopolies through patent and copyright protection is predicated on the assumption that the benefit gained by promoting the sciences and useful arts outweighs the economic costs associated with monopolies.<sup>73</sup> To promote the development of software, legal protection is necessary to overcome the market shortcomings that affect software. The two market shortcomings that primarily affect software are the problems of public goods and externalities.<sup>74</sup> To properly promote software development, courts must first understand the economic balancing that is inherent in the applicable patent and copyright laws; and second, the courts must understand how the problems of public good and externalities affect this economic balance.<sup>75</sup>

Congress determined that economic incentives to authors and inventors are the best way to promote the arts and sciences.<sup>76</sup> This determination is demonstrated by the creation of patent and copyright laws. Congress provided economic incentives by granting limited monopolies that give authors and inventors exclusive rights.<sup>77</sup> Congress granted authors the exclusive right to copy their works,<sup>78</sup> while also giving inventors

---

72. Howard Root, Note, *Copyright Infringement of Computer Programs: A Modification of the Substantial Similarity Test*, 68 MINN. L. REV. 1264, 1292 (1984) ("The computer software industry progresses by a stepping-stone improvement process, with each innovation building on past innovations to produce an improved product.").

73. See ROBERT L. HEILBRONER, UNDERSTANDING MICROECONOMICS 121-24 (2d ed. 1972) (maintaining that compared to competitive markets, monopolies impose two costs on society: hire prices and lower supply).

74. Menell, *supra* note 33, at 1058-59 (noting that public goods problem is major concern with regard to protection for computer programs).

75. For a discussion of this balancing of interests, see *supra* notes 41-46 and accompanying text.

76. Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417, 429 (1984). In *Sony Corp. of America v. Universal City Studios, Inc.*, Justice Stevens recognized that the monopoly privileges granted by Congress are "intended to motivate the creative activity of authors and inventors by the provision of a special reward." *Id.*

77. The Constitution grants Congress the power "[t]o promote the Progress of Science and useful Arts by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries." U.S. CONST. art. I, § 8. However, the Supreme Court has maintained that "[t]he monopoly privileges that Congress may authorize are [not] unlimited." *Sony Corp.*, 464 U.S. at 429.

78. 17 U.S.C. § 106 (1988). The Copyright Act states that: "Copyright protection subsists . . . in original works of authorship fixed in any tangible medium of

the exclusive right to use their inventions.<sup>79</sup> Ultimately, however, the purpose of the copyright and patent laws is to benefit society by increasing the availability of creative works.<sup>80</sup> Therefore, the exclusive rights granted by copyright and patent laws involve a congressionally determined tradeoff between private incentives and social benefits.<sup>81</sup>

expression." 17 U.S.C. § 102(a) (1988). The Act grants the author five exclusive rights:

(1) to reproduce the copyrighted work in copies or phonorecords; (2) to prepare derivative works based upon the copyrighted work; (3) to distribute copies or phonorecords of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending; (4) in the case of literary, musical, dramatic, and choreographic works, pantomimes, and motion pictures and other audiovisual works, to perform the copyrighted work publicly; and (5) in the case of literary, musical, dramatic, and choreographic works, pantomimes, and pictorial, graphic, or sculptural works, including the individual images of a motion picture or other audiovisual work, to display the copyrighted work publicly.

*Id.* § 106.

79. 35 U.S.C. §§ 101-103 (1988). The Patent Act grants a patent to any new and useful process or improvement based upon non-obviousness. *Id.* For a 17 year term, a patent grants the inventor "the right to exclude others from making, using, or selling" the invention or process throughout the United States. 35 U.S.C. § 154 (1988).

80. *United States v. Paramount Pictures*, 334 U.S. 131, 158 (1948) ("The copyright law, like the patent statutes, makes reward to the owner a secondary consideration . . . . It is said that reward to the author or artist serves to induce release to the public of the products of his creative genius.") (citations omitted); *Fox Film Corp. v. Doyal*, 286 U.S. 123, 127 (1932) ("The sole interest of the United States and the primary object in conferring the monopoly lie in the general benefits derived by the public from the labors of authors."). In *Sony Corp.*, the Court discussed Congress' explanation of the Copyright Act's purpose:

The enactment of copyright legislation by Congress under the terms of the Constitution is not based upon any natural right that the author has in his writings, . . . but upon the ground that the welfare of the public will be served and progress of science and useful arts will be promoted by securing to authors for limited periods the exclusive rights to their writings.

*Sony Corp.*, 464 U.S. at 429 n.10 (quoting H.R. REP. NO. 2222, 60th Cong., 2d Sess. 7 (1909)).

81. *Sony Corp.*, 353 U.S. at 429. The text of the Constitution makes it clear that Congress has the responsibility for defining the limited scope of protection for authors and inventors. *Id.* The Court in *Sony Corp.* stated that "this task involves a difficult balance between the interests of authors and inventors in the control and exploitation of their writings and discoveries on the one hand, and society's competing interest in the free flow of ideas, information, and commerce on the other hand . . . ." *Id.* When revising the Copyright Act in 1909, Congress enunciated this balance stating:

In enacting a copyright law Congress must consider . . . two questions: First, how much will the legislation stimulate the producer and so benefit the public; and, second, how much will the monopoly granted be detrimental to the public? The granting of such exclusive rights, under the proper terms and conditions, confers a benefit upon the public that outweighs the evils of the temporary monopoly.

*Id.* at 429 n.10 (quoting H.R. REP. NO. 2222, 60th Cong., 2d Sess. 7 (1909)).

While Congress rewards inventors and authors by granting limited monopolies, the Constitution states that the ultimate aim of legal protection is "[t]o promote the Progress of Science and useful Arts."<sup>82</sup> Given this aim, it would be inconsistent for Congress to protect a work simply because the author expended considerable energy and skill producing it.<sup>83</sup>

Private incentives for authors and inventors are needed because a free-market economy will under-allocate resources directed toward both research and utilization of research.<sup>84</sup> Therefore, the inability of authors to gain a return commensurate with their efforts should be corrected by government-granted monopolies.<sup>85</sup> The two principal shortcomings of the market, which necessitate government intervention for computer applications, are public goods and network externalities.<sup>86</sup>

---

In *Computer Associates*, the Second Circuit noted that copyright law seeks to establish a "delicate equilibrium" between adequate protection to promote creative works and excess protection that allows monopolistic stagnation. *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 696 (2d Cir. 1992). The United States District Court for the Northern District of California followed a similar rationale in *Apple Computer, Inc. v. Microsoft Corp.*, stating that "copyright attempts to maintain [a balance that] prevent[s] free riders from ripping off creative expression while not stifling others from improving or extending that expression." 799 F. Supp. 1006, 1025 (N.D. Cal. 1992).

82. U.S. CONST. art. I, § 8.

83. *Computer Assocs.*, 982 F.2d at 711. The notion that a copyright was a reward for the hard work that went into creating the work was known as the "sweat of the brow" doctrine. *Feist Pubs., Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 352 (1991). In *Feist Publications, Inc. v. Rural Telephone Service Co.*, the Supreme Court ruled that an uncopyrightable work is not made copyrightable simply because it resulted from considerable time and effort. *Id.*

84. FINDING A BALANCE, *supra* note 1, at 184. The CONTU concluded that:

To sum up, we expect a free enterprise economy to underinvest in invention and research (as compared with an ideal) because it is risky, because the product can be appropriated only to a limited extent, and because of increasing returns in use. This underinvestment will be greater for more basic research. Further, to the extent that a firm succeeds in engrossing the economic value of its inventive activity, there will be an underutilization of that information as compared with an ideal allocation.

*Id.* (quoting Kenneth J. Arrow, *Economic Welfare and the Allocation of Resources for Invention*, in NATIONAL BUREAU OF ECONOMIC RESEARCH, *THE RATE AND DIRECTION OF INVENTIVE ACTIVITY: ECONOMIC AND SOCIAL FACTORS* (1962)). CONTU went on to note that "[b]ecause information is intangible, even with legal protections, sellers cannot fully appropriate its value. On the demand side, potential buyers find it difficult to value information correctly, unless they have already acquired it." *Id.* at 184 n.6.

85. *See id.* at 184. The Supreme Court noted that "[s]acrificial days devoted to such creative activities deserve rewards commensurate with the services rendered." *Mazer v. Stein*, 347 U.S. 201, 219 (1954).

86. Menell, *supra* note 33, at 1058-59. These market failures are somewhat offset by market cures. *Id.* at 1060. Market cures may sufficiently promote development of software because the developer recognizes an economic gain. *Id.* Therefore, these cures must be included in the calculation when determining what protection to afford software. *See id.* Consider the following three market

A public good has two characteristics: (1) nonexcludability and (2) nonrivalrous competition.<sup>87</sup> Nonexcludability means that the author is unable to exclude any person from benefitting from the good, regardless of whether the person pays for it.<sup>88</sup> Nonrivalrous competition means that additional consumers of the product do not deplete the supply.<sup>89</sup>

Computer software is a prime example of a public good.<sup>90</sup> First, software is nonexclusive because a software user can copy the software and disseminate it to numerous other users who receive the benefit of the program without paying the developer.<sup>91</sup> Second, because the character-

---

cure examples. First, software innovators can earn a prominent place in the market which their competitors must overcome. *Id.* Second, these software innovators can require long-term maintenance and update contracts, although such a cure is generally not available for high volume application software that is sold to personal computer users. *See id.* Finally, software innovators can install anticopying devices into programs. *Id.* While possible, this market cure option is rarely utilized because users do not like anticopying devices; the devices limit their ability to make back up copies. *Id.*

In addition to these market cures, innovators can implement other tactical measures. For example, software innovators can enter into licensing agreements with users that prohibit reproduction and dissemination. *Id.* at 1060-61. While this is a viable option for software to be used by main frame and minicomputers, it is not clear what effect this has on the software market for personal computers. *Id.* at 1061. Currently, software for personal computers is marketed with "shrink-wrap" licensing agreements. *Id.* A "shrink-wrap" agreement is included with packaged software and "purports to bind the purchaser to the terms of a licensing agreement if the purchaser removes the heat-sealed plastic wrapping surrounding the documentation and/or floppy disk." *Id.* at 1061 n.81. The utility of these agreements is questionable, because their enforceability is uncertain. *Id.* at 1061. Another tactic available to software innovators is the development of research consortia, whereby development costs are spread over a group of innovators. *Id.*

87. *Id.* at 1059. Classic examples of public goods are: lighthouses, television signals, beautiful gardens and national defense. *Id.* To overcome the market distortions, the government produces some public goods, such as national defense and pays for them by collective taxation. FINDING A BALANCE, *supra* note 1, at 185.

88. Menell, *supra* note 33, at 1059.

89. *Id.* Traditional examples of nonrivalrous competition are air, water and empty space in the environment. HEILBRONER, *supra* note 73, at 140. However, because of changes in the environment, these examples have actually lost their "free" characteristics and have become scarce. *Id.*

90. Menell, *supra* note 33, at 1059. The CONTU report recognized that the cost of development is far greater than the cost of duplication, which meant that legal as well as physical protection is required if computer software is to be created and disseminated. FINAL REPORT, *supra* note 15, at 10-11 (1979). According to Peter Menell:

[g]iven the ease and low cost of copying application programs, it is often impossible to exclude nonpurchasers from an application program's benefits once it is commercially available. Moreover, one person's use of the information does not detract from others' use of that same information. Since application program creators cannot reap the marginal value of their efforts under these circumstances, they will, in the absence of other incentives to innovate, tend to undersupply new and better application programs.

Menell, *supra* note 33, at 1059.

91. Menell, *supra* note 33, at 1059.

istic of nonrivalrous competition is inherent in software, unauthorized duplication does not diminish the original user's enjoyment of the software.<sup>92</sup> By granting a limited monopoly, the government decreases the problem associated with public goods, thereby encouraging individuals to produce new and better products.<sup>93</sup>

In addition to the problem of public goods, software is affected by externalities.<sup>94</sup> Externalities are social costs or benefits that affect parties who are not involved in an economic transaction.<sup>95</sup> A prime example of an externality in the computer field is the development of standards.<sup>96</sup> Computer users desire application programs that use standardized computer-human interfaces.<sup>97</sup> When developing software, a programmer has two choices: 1) utilize an emerging or existing standard or 2) create a non-compatible standard.<sup>98</sup> By utilizing a compatible standard, the developer's target market will be larger;<sup>99</sup> however, the developer's market share will be smaller.<sup>100</sup>

Standards benefit computer users in three ways. Standards promote mobility in the workplace, reduce retraining costs and allow a skill to be used in other environments.<sup>101</sup> The standards benefit is reduced, how-

92. *Id.*

93. *Id.* at 1059-60.

94. *Id.* at 1058-59.

95. HEILBRONER, *supra* note 73, at 141. Externalities are costs of producing or consuming goods that bypass the price system. *Id.* Externalities are a pervasive characteristic of the economic process and even a perfect market system cannot fully account for their benefits or costs. *Id.*

An example of externalities is the production and use of automobiles. *Id.* The factory emits smoke that lowers the value of property in the area. *Id.* Additionally, the use of automobiles may require the installation of traffic lights. *Id.* These costs are not included in the private cost to the producer and user. *Id.* Rather, these costs are passed on to society and are paid for by both users and nonusers. *Id.*

96. Menell, *supra* note 33, at 1066. An example of standardization is the standard typewriter keyboard known as the 'QWERTY' keyboard. *Id.* Its adoption as a standard allows everyone to be trained on the same keyboard configuration. *Id.*

97. *Id.*

98. *See id.* at 1066-67.

99. *Id.* at 1067-68. According to Menell,

a firm entering the market enlarges the size of a network comprising both its product and its rivals' products. This will have the effect of increasing the desirability of the rivals' products to consumers, thereby reducing the adopter's market share (although of a larger market) relative to what it would have been had the firm adopted a noncompatible product standard.

*Id.*

100. *Id.*

101. *Id.* at 1066. Emphasizing the benefits of decreased retraining costs, John Sculley stated: "It's becoming apparent that the real cost is not the hardware or the software. The real cost is teaching the user." *Interview: Apple Computer, Inc., President and Chairman John Sculley-On Fitting into the IBM World of Computing*, PERSONAL COMPUTING, Apr. 1986, at 145, 147. An example of skill usage in other

ever, when users incur costs to adopt a lesser efficient standard.<sup>102</sup> These costs may be incurred because a well-established standard "can create excess inertia that makes it much more difficult for any one producer to break away from the old standard."<sup>103</sup> This excess inertia hinders the development and adoption of new and better standards.<sup>104</sup> Consequently, a new standard will only be adopted if the standard has benefits that outweigh the costs of retraining employees and replacing existing hardware and software. In the final analysis, if developers are not given proper incentives to create and improve standards, then standards that are not the most efficient may become entrenched in the marketplace.

The policies that favor protecting computer programs must be balanced against the adverse effect that the protection may have on the market.<sup>105</sup> To determine how this balance should best be struck, Congress established the National Commission on New Technological Uses of Copyrighted Works (CONTU), a committee created to recommend the

environments is the use of keyboard skills in word processing, spreadsheets and databases. Menell, *supra* note 33, at 1066.

102. Menell, *supra* note 33, at 1067.

103. *Id.* at 1070. The 'QWERTY' key board is an example of this. *Id.* at 1067. The 'QWERTY' keyboard's configuration was designed so that the key arms would not entangle. *Id.* This is no longer a concern with computers. *Id.* An alternate keyboard design was patented in 1932, called the Dvorak Simplex Keyboard. Because of its more efficient configuration, the current speed typing record was established using a Dvorak Simplex Keyboard. *Id.* However, because of the widespread use of the 'QWERTY' keyboard, the Dvorak keyboard has not been adopted. *Id.*

104. *Id.* at 1070. "[O]verly inclusive copyright protection can produce its own negative effects by inhibiting the adoption of compatible standards (and reducing so-called 'network externalities'.)" Apple Computer, Inc. v. Microsoft Corp., 700 F. Supp. 1006, 1025 (N.D. Cal. 1992).

105. Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156 (1975). The Court stated:

The limited scope of the copyright holder's statutory monopoly, like the limited copyright duration required by the Constitution, reflects a balance of competing claims upon the public interest: Creative work is to be encouraged and rewarded, but private motivation must ultimately serve the cause of promoting broad public availability of literature, music, and other arts. The immediate effect of our copyright law is to secure a fair return for an "author's" creative labor. But the ultimate aim is, by this incentive, to stimulate artistic creativity for the general public good. . . . When technological change has rendered its literal terms ambiguous, the Copyright Act must be construed in light of this basic purpose.

*Id.* (citations omitted). This balance has been recognized for several centuries. In 1785, Lord Mansfield noted that

we must take care to guard against two extremes equally prejudicial; the one, that men of ability, who have employed their time for the service of the community, may not be deprived of their just merits, and the reward of their ingenuity and labour; the other, that the world may not be deprived of improvements, nor the progress of the arts be retarded.

Cary v. Longman, 102 Eng. Rep. 138, 140 n.6 (1801) (quoting Sayre v. Moore (1785)).

best form of protection for computer software.<sup>106</sup> CONTU concluded that as between copyright, patent, and trade secret law, copyright law would have the smallest negative impact on the market.<sup>107</sup> As a result of CONTU, Congress amended the Copyright Act, and now, at a minimum, copyright law protects the literal code of computer programs.<sup>108</sup>

Recently, courts have also been faced with the issue of determining the level of protection to afford the nonliteral aspect of computer software.<sup>109</sup> In confronting the issue, courts have sought a systematic approach to determining the scope of protection for nonliteral aspects of computer software. Rather than following an inflexible formula for estab-

---

106. See FINAL REPORT, *supra* note 15. Because Congress adopted CONTU's recommendations without any alterations or committee reports, some courts have concluded that CONTU's report should be regarded as legislative history. *Wheilan Assocs. v. Jaslow Dental Lab.*, 797 F.2d 1222, 1241 (3d Cir. 1986) (citing *Micro Spare, Inc. v. Amtype Corp.*, 592 F. Supp. 33, 35 n.7 (D. Mass. 1984) and *Midway Manufacturing Co. v. Strohon*, 564 F. Supp. 741, 750 n.6 (N.D. Ill. 1983)).

107. FINAL REPORT, *supra* note 15, at 18. CONTU stated that both patent and trade secret protection "may inhibit the dissemination of information and restrict competition to a greater extent than copyright." *Id.* at 16. The report noted that "[p]atents are designed to give investors a short-term, powerful monopoly in devices, processes, compositions of matter, and designs which embody their idea." *Id.*

Trade secret protection on the other hand, has several high costs. *Id.* at 17. Maintaining adequate security can be very expensive. *Id.* In fact, maintaining secrecy with respect to widely disseminated works may be virtually impossible. *Id.* This "substantially precludes the use of trade secrecy with respect to programs sold in multiple copies over the counter to small business, schools, consumers, and hobbyists." *Id.* "Experts in the computer industry state that a further problem with respect to trade secrecy is that there is much human effort wasted when people do for themselves that which others have already done but are keeping secret." *Id.* at 17-18.

108. *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 702 (2d Cir. 1992) ("It is now well settled that the literal elements of computer programs, i.e., their source code and codes, are the subject of copyright protection."). Protection of the literal elements of computer programs is necessary because "[o]therwise, an imitator could immediately enter the first programmer's market at much lower product development costs. This prospect would discourage innovation and reward waiting." Menell, *supra* note 33, at 1080. For an explanation of source code and object code see *supra* note 66.

109. See, e.g., *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992); *CMAX/Cleveland, Inc. v. UCR, Inc.*, 804 F. Supp. 337 (M.D. Ga. 1992) (infringement for program's file structures, transaction codes, screen and reports); *Apple Computer Inc. v. Microsoft Corp.*, 799 F. Supp. 1006 (N.D. Cal. 1992) (alleging infringement of graphical user interface); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F. Supp. 203 (D. Mass. 1992) (infringing upon user interface for spreadsheet program); *Gates Rubber Co. v. Bando Am., Inc.*, 798 F. Supp. 1499 (D. Colo. 1992) (claiming infringement upon structure of program used to calculate industrial belt configurations); *Softel, Inc. v. Dragon Medical & Scientific Communications, Ltd.*, No. 87-0167, 1992 WL 168190 (S.D.N.Y. June 30, 1992) (claiming infringement for non-literal elements used in interactive videotape program).



lishing the scope of copyright protection, these courts should base their decisions on the balance of interests inherent in the Copyright Act.<sup>110</sup>

D. *The Third Circuit's Approach in Whelan Associates v. Jaslow Dental Laboratories*

Prior to *Computer Associates*, the United States Court of Appeals for the Third Circuit in *Whelan Associates, Inc. v. Jaslow Dental Laboratory Inc.*<sup>111</sup> was the only court to determine whether copyright protection extends to the non-literal elements of a computer program.<sup>112</sup> While other courts have made reference to the non-protectability of non-literal program elements,<sup>113</sup> none of these courts have been confronted with the issue of protectability of non-literal programs.<sup>114</sup>

*Whelan* raised the issue of whether there were significant similarities between two computer programs—Dentalab and Dentcom.<sup>115</sup> Elaine Whelan created a software package called Dentalab for the defendant, Jaslow Dental Laboratories (Jaslow).<sup>116</sup> Whelan Associates negotiated a royalty agreement with Jaslow under which Jaslow received a commission

---

110. In *Apple Computer's* recent suit, the district court noted that "copyright attempts to maintain [a balance of] preventing free riders from ripping off creative expression while not stifling others from improving or extending that expression. *Apple Computer*, 799 F. Supp. at 1025. Additionally, in *Computer Assocs.*, the Second Circuit stated that copyright law seeks to establish a "delicate equilibrium" between adequate protection to promote creative works, and excess protection that allows monopolistic stagnation. *Computer Assocs.*, 982 F.2d at 696. For a discussion of the balancing of interests, see *supra* notes 73-83 and accompanying text.

111. 797 F.2d 1222 (3d Cir. 1986).

112. *Computer Assocs.*, 982 F.2d at 705.

113. See generally *Johnson Controls, Inc. v. Phoenix Control Sys.*, 886 F.2d 1173, 1175 (9th Cir. 1989) (upholding preliminary injunction; court noted that copyright for computer program may be infringed without copying its literal elements); *Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv.*, 807 F.2d 1256, 1262 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987) (finding no abuse of discretion by district court in denying injunction for copying program structure and avoiding formulation of new rule); *Digital Communications Assocs. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 460 (N.D. Ga. 1987) (deciding to protect "the arrangement of the status screen [because it] involve[d] considerable stylistic creativity"); *Broderbund Software, Inc. v. Unison World, Inc.*, 648 F. Supp. 1127, 1133 (N.D. Cal. 1986) (holding that copyright protection existed for "overall structure, sequencing, and arrangement of screens" in plaintiff's program).

114. *Computer Assocs.*, 982 F.2d at 705 ("In the context of computer programs, the Third Circuit's noted decision in *Whelan* has, thus far, been the most thoughtful attempt to [distinguish protectable expression from ideas].").

115. *Whelan*, 797 F.2d at 1233.

116. *Id.* at 1225-26. When she created the software, Ms. Whelan was employed by a software development company, Strohl Systems Group, Inc. ("Strohl"). *Id.* at 1225. Under the terms of their contract, Strohl retained the rights to Dentalab, and Jaslow was paid a 10% royalty on all sales of Dentalab. *Id.* Ms. Whelan acquired the rights to Dentalab from Strohl, and formed Whelan Associates. *Id.* at 1226.

for each copy of Dentalab that was sold.<sup>117</sup> After three years, Jaslow terminated the agreement with Whelan Associates<sup>118</sup> and attempted to claim sole rights to the Dentalab software.<sup>119</sup> After terminating its agreement with Whelan Associates, Jaslow employed a professional programmer to write Dentcom, which was a version of Dentalab in a different computer language.<sup>120</sup> Because Dentcom was written in a different computer language, there were virtually no literal similarities between Dentalab and Dentcom.<sup>121</sup> However, the new version had a structure that was virtually identical to Dentalab.<sup>122</sup>

In determining the extent of copyright protection for Dentalab's non-literal elements, the Third Circuit focused its attention on the considerations underlying copyright law.<sup>123</sup> The court discussed the idea/expression distinction, and the limitations of merger and *scènes à faire*.<sup>124</sup> From this discussion, the Third Circuit formulated a two-step approach for determining the scope of copyright protection for computer programs. First, a court must identify the function or purpose of the program,<sup>125</sup> also referred to as the program's idea.<sup>126</sup> Second, a court must determine if the particular expression used is necessary to the program's idea.<sup>127</sup> If it is not necessary, then the expression is protectable.<sup>128</sup>

In applying the test, the Third Circuit stated that the idea behind Dentalab was the efficient organization of a dental laboratory.<sup>129</sup> Because there were a variety of program structures through which the idea could have been expressed, the structure was expression.<sup>130</sup> Therefore, the

117. *Id.*

118. *Id.* The original agreement between the parties was for one year and was then terminable at will by either party on 30 days notice. *Id.*

119. *Id.*

120. *Id.* at 1227. The Third Circuit noted that Edward Jaslow, a Jaslow employee, "surreptitiously and without consent of . . . Whelan Associates obtained a copy of the source code." *Id.* at 1232 n.22.

121. *Id.* at 1233 (noting that district court found no similarity between literal elements).

122. *Id.* at 1242-48. For an illustration of the similarities between the structure of the order entry module, see CLAPES, *supra* note 3, at 106-07.

123. *Whelan*, 797 F.2d at 1235 ("[P]recisely because the line between idea and expression is elusive, we must pay particular attention to the pragmatic considerations that underlie the distinction and copyright law generally.>").

124. *Id.* at 1235-36.

125. *Id.* at 1236.

126. *Id.* ("[T]he purpose or function of a utilitarian work would be the work's idea.") (emphasis omitted).

127. *Id.*

128. *Id.* ("Where there are various means of achieving the desired purpose, then the particular means chosen is not necessary to the purpose; hence, there is expression, not idea.").

129. *Id.* at 1240.

130. *Id.*

Third Circuit concluded that the non-literal elements of Dentalab were protected by copyright.<sup>131</sup>

Several commentators have criticized the Third Circuit's decision as affording overly broad protection for software.<sup>132</sup> One criticism is that the Third Circuit in *Whelan* misunderstood the nature of a computer program. As such, by holding that a program has only one idea, the Third Circuit did not realize that a computer program consists of numerous smaller programs, called subprograms, each of which has its own idea.<sup>133</sup> The United States Court of Appeals for the Second Circuit agreed with this criticism of the *Whelan* approach, and formulated a different approach in *Computer Associates International, Inc. v. Altai, Inc.*<sup>134</sup>

### III. *COMPUTER ASSOCIATES INTERNATIONAL, INC. v. ALTAI, INC.*

#### A. *Factual Background*

In *Computer Associates International, Inc. v. Altai, Inc.*, Computer Associates sued Altai for copyright infringement, alleging that Altai had copied both literal and non-literal elements of one of Computer Associates' programs.<sup>135</sup> The United States District Court for the Eastern District of New York upheld Computer Associates' claim of literal infringement, but it denied the claims of non-literal infringement.<sup>136</sup>

The allegedly infringed work at issue was Computer Associates' CA-SCHEDULER, a job scheduling program designed to operate on IBM mainframe computers.<sup>137</sup> The program had an operating system compatibility program named ADAPTER.<sup>138</sup> This element of the program received commands from the main program and interacted with the

131. *Id.*

132. NIMMER, *supra* note 16, § 13.03[F], at 13-78.33 (concluding that *Whelan's* crucial flaw is that it assumes only one idea underlies any program, and once separable idea is identified, everything else must be expression); Menell, *supra* note 33, at 1082 (maintaining that monopoly power bestowed by *Whelan* rule is not justified on basis of sound public policy analysis); Englund, *supra* note 15, at 881 (observing that *Whelan* rule may make it impossible for others to create an efficient program that will perform same function as original program); Marc T. Kretschmer, Note, *Copyright Protection for Software Architecture: Just Say No!*, 1988 COLUM. BUS. L. REV. 823, 837-39 (arguing that *Whelan* court did not meet challenge of finding last idea that was still distinct from expression).

133. Kretschmer, *supra* note 132, at 839 (stating that court treated program as single unified expression rather than examining each of program's multiple facets).

134. 982 F.2d 693 (2d Cir. 1992).

135. 775 F. Supp. 544 (E.D.N.Y. 1991), *aff'd*, 982 F.2d 693 (2d Cir. 1992).

136. *Id.* at 562.

137. *Computer Assocs.*, 982 F.2d at 698. A scheduling program specifies when a computer should perform various tasks. *Id.* It also controls the computer as it executes the schedule. *Id.*

138. *Id.*

computer's operating system.<sup>139</sup> ADAPTER allowed CA-SCHEDULER to be used by three different operating systems.<sup>140</sup> Without ADAPTER, different versions of CA-SCHEDULER would have been required for each different operating system.<sup>141</sup> Moreover, separate marketing strategies would have been necessary for each version, while individual changes to each version would have been necessary.<sup>142</sup>

The president of Altai, James Williams, developed ZEKE, a competitor of CA-SCHEDULER.<sup>143</sup> The initial version of ZEKE was only capable of running on one operating system<sup>144</sup> because it did not have a separate compatibility component such as ADAPTER. To improve ZEKE, Williams hired Claude F. Arney who was employed by Computer Associates at the time.<sup>145</sup> Arney informed Williams of the compatibility component used in CA-SCHEDULER.<sup>146</sup> Arney and Williams decided to rewrite ZEKE and incorporate a compatibility component.<sup>147</sup>

Arney wrote the compatibility component for ZEKE,<sup>148</sup> entitling it OSCAR 3.4.<sup>149</sup> While writing OSCAR 3.4, Arney referred to a copy of CA-SCHEDULER's source code that he had illicitly retained after leaving Computer Associates.<sup>150</sup> As a result, approximately thirty percent of OSCAR's source code was copied from CA-SCHEDULER.<sup>151</sup> No one else at

139. *Id.* Operating systems are designed to manage the resources of the computer and allocate those resources to other programs within the computer. *Computer Assocs.*, 775 F. Supp. at 549-50. Examples of functions performed by an operating system are: "[C]hanneling information entered at a keyboard to the proper application program; sending information from an application program to a display screen; providing blocks of memory to an application program . . . ." *Id.* at 550.

140. *Computer Assocs.*, 982 F.2d at 698. CA-SCHEDULER is separated into two parts: One that contains task-specific operations that are independent of the operating system, and one that contains all of the operations that interact with the operating system. *Id.* at 699. The second part can be written so that requests received from the first part can be understood by whatever operating system is being used by the computer. *Id.* In this way a single program can be understood by numerous operating systems. *Id.*

141. *Id.* at 699.

142. *Id.*

143. *Id.*

144. *Id.* ZEKE was designed to be used with IBM's VSE operating system. *Id.*

145. *Id.* Williams and Arney were long-standing friends, and had worked together at Computer Associates. *Id.*

146. *Id.* When Williams asked Arney to work for Altai, Williams did not know that ADAPTER was an element of CA-SCHEDULER. *Id.* However, Arney was "intimately familiar with various aspects of ADAPTER." *Id.*

147. *Id.* at 700. The increased compatibility of a component such as ADAPTER makes a program "significantly more marketable." *Id.* at 699.

148. *Id.* at 700. It took Arney four months to rewrite the program. *Id.*

149. *Id.*

150. *Id.* The district court noted that Arney recognized that taking the source code was "contrary to the agreements he had signed prohibiting employees from retaining such copies." *Id.*

151. *Id.*

Altai knew that Arney had used CA-SCHEDULER's source code until Computer Associates filed suit against them.<sup>152</sup>

After Computer Associates commenced their suit, Altai began a lengthy process of rewriting OSCAR 3.4.<sup>153</sup> Williams identified the copied sections from CA-SCHEDULER<sup>154</sup> and assembled a team of eight Altai programmers who had not been involved in writing or improving OSCAR 3.4.<sup>155</sup> Each programmer was assigned a segment of the program.<sup>156</sup> During this process, the CA-SCHEDULER source code was locked away, and the programmers were forbidden to talk to Arney about OSCAR.<sup>157</sup> Once completed, the new program was sent to all new customers.<sup>158</sup> It was also sent as a "free upgrade" to all of the customers using OSCAR 3.4.<sup>159</sup>

The district court held that OSCAR 3.4 had infringed the literal elements of CA-SCHEDULER, but OSCAR 3.5 did not infringe either the literal or non-literal aspects of CA-SCHEDULER.<sup>160</sup> On appeal, Computer Associates maintained two contentions: (1) the district court applied an erroneous method for determining substantial similarity between the computer programs and (2) the district court's test failed to account for computer program's non-literal elements.<sup>161</sup> On appeal, Altai conceded the damage award for the literal copying of ADAPTER into OSCAR 3.4.<sup>162</sup> Because there was no dispute that Computer Associates owned a valid copyright in CA-SCHEDULER 2.1, which contained ADAPTER, the only question left for the Second Circuit to resolve was whether there was substantial similarity between the non-literal aspects of OSCAR 3.5 and ADAPTER.<sup>163</sup>

152. *Id.*

153. *Id.* The process took eight programmers almost six months to complete. *Id.*

154. *Id.* Williams never actually looked at the ADAPTER source code. *Id.* Rather, Arney told him which sections of OSCAR were taken from ADAPTER. *Id.*

155. *Id.*

156. *Id.* Williams provided each programmer with a description of ZEKE's functions so that they could rewrite the code. *Id.*

157. *Id.* For purposes of analysis, both the district court and the Second Circuit assumed that Altai had access to ADAPTER during the rewriting process. *Id.* at 701. However, it is apparent from the court's decision that it did not believe that Altai used the ADAPTER code during the rewriting process. *See id.* at 700.

158. *Id.*

159. *Id.*

160. *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 775 F. Supp. 544, 562 (E.D.N.Y. 1991), *aff'd*, 982 F.2d 693 (2d Cir. 1992).

161. *Computer Assocs.*, 982 F.2d at 701. Computer Associates also asserted that the district court improperly concluded that its state trade secret claim was preempted by the Copyright Act. *Id.* The Second Circuit upheld the district court's holding on both points of appeal. *Id.* at 721.

162. *Id.* at 701.

163. *Id.* Computer Associates also appealed the denial of their trade secret claim. *Id.* The Second Circuit held that the federal copyright law preempted the trade secret claim. *Id.* at 715-21.

B. *Case Analysis*

To determine the scope of copyright protection for a computer program's literal aspects, the Second Circuit developed the Abstraction-Filtration test.<sup>164</sup> The Second Circuit found it necessary to develop this test after concluding that the test previously developed by the Third Circuit in *Whelan* afforded overly-broad protection.<sup>165</sup> The goal of the Abstraction-Filtration test has been to determine whether the non-literal aspects of two programs are substantially similar.<sup>166</sup>

Before addressing the issue of substantial similarity, the Second Circuit focused on three preliminary issues: (1) whether Computer Associates had a valid copyright; (2) whether Altai had access to the allegedly infringed program;<sup>167</sup> and (3) whether the non-literal aspects of computer programs are protectable.<sup>168</sup> The first issue regarding the validity of Computer Associate's Copyright was essentially a moot point because Computer Associates had registered their copyright for ADAPTER.<sup>169</sup> The second issue—Altai's access to the allegedly infringed program—was also resolved in Computer Associate's favor because the Second Circuit followed the assumption that Altai had access to ADAPTER.<sup>170</sup>

With respect to the third issue—determining whether non-literal aspects of programs are protected—the Second Circuit was faced with a compelling syllogism: If computer programs are literary works and the non-literal aspects of literary works are protected, then the non-literal aspects of computer programs should likewise be protected.<sup>171</sup> While the Second Circuit believed that non-literal aspects of computer programs should be protected, it noted that the scope of the protection must be determined.<sup>172</sup>

164. *Id.* at 706-12.

165. *Id.* at 706 (noting that *Whelan* decision relied too heavily on "metaphysical distinctions" and did not place enough emphasis on practical considerations).

166. *Id.*

167. *Id.* at 701. For a discussion of the requirements for proving copyright infringement, see *supra* notes 34-35 and accompanying text.

168. *Computer Assocs.*, 982 F.2d at 702.

169. *Computer Assocs. Int'l v. Altai, Inc.*, 775 F. Supp. 544, 555 (E.D.N.Y. 1991), *aff'd*, 982 F.2d 693 (2d Cir. 1992). For a discussion of the effect of registering a copyright, see *supra* note 34.

170. *Computer Assocs.*, 982 F.2d at 701.

171. *Id.* at 702. Congress has defined computer programs as literary works. 17 U.S.C. § 102(b) (1988). For a discussion of the non-literal protection of literary works, see *supra* notes 36-39 and accompanying text.

172. *Computer Assocs.*, 982 F.2d at 702-03. ("We have no reservation in joining the company of those courts that have already ascribed to [the] logic of [non-literal protection for software] . . . . However, that conclusion does not end our analysis.") (citations omitted).

Recognizing the existence of certain inherent conflicts, the Second Circuit created the Abstraction-Filtration test.<sup>173</sup> The Abstraction-Filtration test takes several important factors into consideration. First, the test is responsive to the scope limitations imposed by the idea/expression distinction and the merger doctrine.<sup>174</sup> Second, the test accounts for the fact that *scènes à faire* and aspects that were already in the public domain are excluded from protection.<sup>175</sup> Finally, the Abstraction-Filtration test considers these two limitations in light of the various public policy considerations that impinge upon copyright protection.

The Abstraction-Filtration test has three steps: abstraction, filtration and comparison.<sup>176</sup> A court determines where to draw the line between idea and expression by using the abstraction test.<sup>177</sup> The filtration step limits the scope of expression determined by the abstraction.<sup>178</sup> Finally, the protectable aspects of the copyrighted program are compared to the allegedly infringing program.<sup>179</sup>

In applying the abstraction test to computer programs, a court must determine the point at which all of the ideas that underlie a program become sufficiently delineated to warrant protection.<sup>180</sup> The development process of a program offers divisions that may correlate with the levels of abstraction.<sup>181</sup> Therefore, to find the level of abstraction, a court should reverse the steps taken in writing a program.<sup>182</sup> The court should start with the code and end with an articulation of the program's ultimate function.<sup>183</sup>

When addressing the level of abstraction, the Second Circuit in *Computer Associates* deferred to the abstraction analysis used by the district court.<sup>184</sup> While the district court determined that the levels of abstrac-

---

173. *Id.* at 706-12. For a discussion of the policy considerations underlying the Copyright Act's balancing of interests, see *supra* notes 73-83 and accompanying text.

174. *Computer Assocs.*, 982 F.2d at 703 ("It is a fundamental principle of copyright law that a copyright does not protect an idea, but only the expression of the idea.").

175. *Id.* at 709-10. For a discussion regarding the way the factors were considered in the Abstraction-Filtration test, see *infra* notes 191-205 and accompanying text.

176. *Computer Assocs.*, 982 F.2d at 706-12.

177. *Id.* at 706-07.

178. *Id.* at 707. The "successive filtering method" embraced by the court was suggested by Professor Nimmer. *Id.* (citing 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03[F] (1992)).

179. *Computer Assocs.*, 982 F.2d at 710 (noting that court must assess "copied portion's relative importance with respect to plaintiff's overall program").

180. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.34.

181. *Id.* at 13-78.33.

182. *Computer Assocs.*, 982 F.2d at 707 (describing example of abstraction for computer program).

183. *Id.*

184. *Id.* at 714. The district court determined the protectable elements of OSCAR 3.5 rather than ADAPTER. *Id.* While the Second Circuit noted that the

tion progressed from the code of the program to the parameter lists and finally to the services required in the general outline, the court did not enunciate the idea or ideas that were the basis for protectable expression.<sup>185</sup>

After the abstraction step is completed, the filtering step is used to remove elements from the scope of copyright protection.<sup>186</sup> The process of filtering removes aspects of the computer program that are efficient, necessary or already in the public domain.<sup>187</sup> Using the doctrine of merger, any aspect that is dictated by efficiency concerns is not protectable.<sup>188</sup> Under the notion of *scènes à faire*, aspects that are necessary, due to constraints by externalities, are not protectable.<sup>189</sup>

The Second Circuit applied the merger doctrine to software, and filtered out aspects of the program that were capable of being expressed in essentially only one way.<sup>190</sup> The court noted that when efficiency con-

---

district court judge should have evaluated the ADAPTER program, "the most serious charge that can be leveled against him is that he was overly thorough in his examination." *Id.*

185. After the rewriting, "there remained virtually no lines of code that were identical to ADAPTER." *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 775 F. Supp. 544, 561 (E.D.N.Y. 1991), *aff'd*, 982 F.2d 693 (2d Cir. 1992). With respect to the parameter lists, "only a few of the lists and macros were similar to protected elements in ADAPTER; the others were either in the public domain or dictated by the functional demands of the program." *Id.* at 562. Finally, the list of services required was not protected because of software compatibility concerns. *Id.*

186. *Computer Assocs.*, 982 F.2d at 707.

187. *Id.* at 707-10.

188. For a discussion of the merger doctrine in the software context, see *infra* notes 190-92 and accompanying text.

189. For a discussion of the *scènes à faire* doctrine in the software context, see *infra* notes 193-203 and accompanying text.

190. *Computer Assocs.*, 982 F.2d at 705 ("[E]lements of a computer program that are necessarily incidental to its function are . . . unprotectable."); *Whelan Assocs. v. Jaslow Dental Lab.*, 797 F.2d 1222, 1236 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) ("Where there are various means of achieving the desired purpose, then the particular means chosen is not necessary to the purpose; hence, there is expression, not idea."); *Softel, Inc. v. Dragon Medical & Scientific Communications, Inc.*, No. 87-0167, 1992 WL 168190, at \*18 (S.D.N.Y. June 30, 1992) ("[W]ith respect to computer programs, the issue is whether a particular idea is capable of being expressed in a different way.") (citations omitted); *Apple Computer Inc. v. Microsoft Corp.*, 799 F. Supp. 1006, 1034-35 (N.D. Cal. 1992) (using merger to limit protection for computer-user interface and holding that folder icon representation was not protectable because expression flowed naturally from underlying idea); *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 66 (D. Mass. 1990) (applying merger doctrine to deny copyright protection for basic spread sheet screen display and "the designation of a particular key that . . . will invoke the menu command system"); *Manufacturers Technologies, Inc. v. Cams, Inc.*, 706 F. Supp. 984, 995-99 (D. Conn. 1989) (denying protection because "[t]he process or manner of navigating internally on any specific screen displays . . . is limited in the number of ways it may be simply achieved to facilitate user comfort"); *Digital Communications Assocs. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 460 (N.D. Ga. 1987) (finding no merger because "modes of expression chosen . . . are clearly not necessary to the idea of the status screen"); *Broderbund*



cerns limit the choices of expression, the merger doctrine applies, and copyright protection is precluded.<sup>191</sup> While in theory there may be numerous ways to write a program to perform a particular function, efficiency constraints may eliminate most alternatives so that there are only one or two workable options.<sup>192</sup>

After applying the merger doctrine, the Second Circuit applied the doctrine of *scènes à faire* to computer programs.<sup>193</sup> It reasoned that an allegedly infringing computer program often contains elements that are not original to the computer program allegedly copied.<sup>194</sup> Rather than flowing from the original author's creativity, the elements are dictated by numerous external considerations and constraints.<sup>195</sup> The court recognized that computer programmers' freedom of expression is limited by several external factors:<sup>196</sup> (1) mechanical specifications of the computer

---

Software v. Unison World, Inc., 648 F. Supp. 1127, 1131-33 (N.D. Cal. 1986); E.F. Johnson Co. v. Uniden Corp., 623 F. Supp. 1485, 1500-02 (D. Minn. 1985).

Additionally, CONTU stated that: "copyrighted language may be copied without infringing when there is but a limited number of ways to express a given idea . . . . In the computer context, this means that when specific instructions, even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to infringement." FINAL REPORT, *supra* note 15, at 20.

191. *Computer Assocs.*, 982 F.2d at 708 (assuming use of particular set of modules is efficiently necessary, then expression has merged with idea of modules and is unprotected); *Softel*, 1992 WL 168190, at \*25 (concluding that "expression which is standard or the most efficient means of accomplishing a task merges with the idea and is not entitled to copyright protection").

Nimmer argues that "[t]he fact that two programs both use the most efficient sorting or searching method available supports an inference of independent creation as readily as it supports one of copying, and thus is not reliable evidence that copying actually occurred." NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.36.

192. *Computer Assocs.*, 982 F.2d at 708; *see also* NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.35 ("Although theoretically many ways may exist to implement a particular idea, efficiency concerns can make one or two choices so compelling as to virtually eliminate any other form of expression.").

193. *Computer Assocs.*, 982 F.2d at 709-10.

194. *Id.*; *see also* Atari Games Corp. v. Oman, 888 F.2d 878, 886 (D.C. Cir. 1989) (discussing *scènes à faire* doctrine in context of limiting copyright protection); Apple Computer Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1035 (N.D. Cal. 1992) (applying *scènes à faire* to exclude copyright protection regarding method of overlapping windows in computer-user interface); Q-Co Indus. v. Hoffman, 625 F. Supp. 608, 616 (S.D.N.Y. 1985) (noting that certain modules were inherent part of any prompting program).

195. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.37; *see also* Plains Cotton Coop. Ass'n v. Good Pasture Computer Serv., Inc., 807 F.2d 1256, 1262 (5th Cir.) (noting that similarities between computer programs were dictated by externalities of cotton market), *cert. denied*, 484 U.S. 821 (1987).

196. *Computer Assocs.*, 982 F.2d at 709 (citing 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.03[F], at 13-65 (1992)). However, Nimmer cautions courts to note that "[c]omputer programming is a highly creative and individualistic endeavor. A court should not be led by defense counsel to believe that complex programs consist only of commonly known techniques and materi-

on which the software will be used;<sup>197</sup> (2) compatibility requirements of other programs with which the program is to be used;<sup>198</sup> (3) computer manufacturers' design standards;<sup>199</sup> (4) demands of target markets;<sup>200</sup>

als strung together without significant originality or skill." NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-79.

197. *Apple Computer*, 799 F. Supp. at 1033 (limiting copyright protection for computer-user interface partly because of hardware constraints); *Lotus Dev. Corp. v. Borland Int'l Inc.*, 799 F. Supp. 203, 213 (D. Mass. 1992) ("[A] program designed to fit hardware specifications cannot be protected by copyright unless the program contains expressive elements not substantially dictated by the hardware."); *Manufacturers Technologies, Inc. v. Cams, Inc.*, 706 F. Supp. 984, 995 (D. Conn. 1989) (denying copyright protection to certain aspects of program because it was "influenced by the type of hardware that the software [was] designed to be used on"); *NEC Corp. v. Intel Corp.*, 1989 Copy. L. Dec. (CCH) ¶ 26,379 (N.D. Cal. 1989) (denying protection partly because expression was constrained by hardware); see also NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.37 (noting that hardware forces programmer to meet machine's design standards; therefore two independent programs designed to be executed by same hardware may contain numerous similarities.).

198. *Lotus Dev.*, 799 F. Supp. at 213. "[A] program designed to interact with preexisting software, such as [an] operating system . . . , is not entitled to protection to the extent that it is constrained by the need for compatibility with the preexisting software." *Id.* There must be proof "that what the program was designed to fit was already in existence before the program was designed to fit it." *Id.*

Two examples of software constraints are the programming language and the operating system. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.40. A programming language will dictate "mechanisms for controlling the flow of data and the ordering of subroutines in the program." *Id.* The operating system chosen will affect how the program makes calls to the actual computer. *Id.*

199. Often computer manufacturers establish standards so that users can "expect a certain pattern in software for a particular machine." NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.40. For example, IBM's *Common User Access SAA Manual* was designed to provide compatibility between IBM PS/2 personal computers, minicomputers and mainframes. *Id.* at 13-78.41. The manual explains what many of the aesthetics of the user interface should look like. *Id.*

200. In *Apple Computer*, the district court concluded that protection should be denied "if 'market factors play a significant factor in determining the sequence and organization' of a computer program." *Apple Computer*, 799 F. Supp. at 1023 (quoting *Plains Cotton Cooperative Association v. Good Pasture Computer Service, Inc.*, 807 F.2d 1256, 1262 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987)). In *Plains Cotton Cooperative Ass'n v. Good Pasture Computer Service, Inc.*, the Fifth Circuit upheld a preliminary injunction because the similarities between two programs were "dictated by the externalities of the cotton market." 807 F.2d 1256, 1262 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987).

Often the practices of the end user will influence a program's design. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.42. As a result, there will be numerous similarities between the two programs designed for the same purpose and the same target market. *Id.* Additionally, in many areas of the software industry, independent developers will compete for the same market. *Id.* at 13-78.28. This means that both will develop software to address the same needs of the same users. *Id.* Therefore, two programs that were created independently could have numerous similarities. *Id.*

A user constraint that is becoming important for all programmers is the general computer-human interaction. This interaction has become a high priority because of the increased number of users who do not have training in computer

and (5) widely accepted programming practices.<sup>201</sup> As such, these elements follow naturally from the work's theme, rather than from the author's creativity,<sup>202</sup> and are subject to filtrations as unprotectable *scènes à faire*.<sup>203</sup> Ultimately, in filtering out material that already existed in the public domain,<sup>204</sup> the Second Circuit found little basis for establishing a software exception to the general rule that works in the public domain are unprotectable.<sup>205</sup>

After all of the unprotectable elements of a work have been filtered out, there remains a core of protectable expression. Thus, what remains for a court to apply is the final step of the Abstraction-Filtration test: the comparison of the protectable aspects of the original work to the allegedly infringing work.<sup>206</sup> In making this comparison, the Second Circuit noted that a quantitatively small misappropriation may infringe a qualitatively vital aspect of the plaintiff's expression.<sup>207</sup> Comparing ADAPTER to OSCAR 3.5, the Second Circuit concluded that Altai had not misappropriated any of ADAPTER's non-literal aspects that were protectable.<sup>208</sup>

---

science. Menell, *supra* note 33, at 1052-53. The computer-human field has identified five goals that a programmer should strive to achieve when designing software: (1) minimize learning time; (2) maximize performance speed; (3) minimize users' rate of errors; (4) maximize user satisfaction; and (5) maximize users' retention of knowledge over time. *Id.* at 1053-54.

201. *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465, 1473 (9th Cir. 1992). The Ninth Circuit noted that "[p]laintiffs may not claim copyright protection of an . . . expression that is, if not standard, then commonplace in the computer software industry." *Id.* (citations omitted). In *Apple Computer*, the district court denied copyright protection to certain aspects of a computer-user interface, partly because it was a standard industry feature. *Apple Computer*, 799 F. Supp. at 1034-36.

There are guidelines for good programming that have developed in the software engineering field. Menell, *supra* note 33, at 1056. These guidelines standardize the design choices in programming regarding substance, structure and form. *Id.* The guidelines are as much a part of the computer industry as are common themes referred to in the literary *scènes à faire*, and therefore, should be denied copyright protection. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-79.

202. NIMMER ON COPYRIGHT, *supra* note 16, § 13.03[F], at 13-78.36.

203. *Id.*

204. *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 710 (2d Cir. 1992).

205. *Id.* ("We see no reason to make an exception to this rule for elements of a computer program that have entered the public domain by virtue of freely accessible program exchanges and the like.")

206. *Id.* ("At this point, the court's substantial similarity inquiry focuses on whether the defendant copied any aspect of [the] protected expression . . .").

207. *Id.* at 714.

208. *Id.*

## IV. CRITICAL ANALYSIS

A. *The Abstraction-Filtration Test's Inherent Discretion*

The Abstraction-Filtration test furthers the policies underlying the copyright laws.<sup>209</sup> The test excludes significant amounts of information that could be useful to the general public.<sup>210</sup> At the same time, the test protects certain aspects of a program's structure.<sup>211</sup> Such protection is an economic incentive that will promote the development of new software.<sup>212</sup>

Perhaps the greatest attribute of the Abstraction-Filtration test is its flexibility. At first blush, it may seem that the test provides negligible protection for a program's non-literal elements. However, the test grants significant discretion to the courts. The abstraction test is inherently discretionary because the line separating an idea from an expression can be drawn anywhere along a continuum.<sup>213</sup> As the line is drawn closer to the program's ultimate function or purpose, the amount of protection will increase. For example, an idea may be defined as a program to organize a dental laboratory. Using the definition, the test will not filter out much of this idea's expression. The idea may alternatively be defined as a program to organize a dental laboratory, utilizing an IBM personal computer. By this definition, certain aspects of the expression will be filtered out because of the hardware's constraints. These examples illustrate how a court can use its discretion to afford the appropriate protection in any particular case.

B. *The Abstraction-Filtration Test Compared to the Whelan Approach*

Applying the Abstraction-Filtration test to the *Whelan* case may be difficult because the Third Circuit in *Whelan* did not enunciate many of the factual matters to be considered when filtering.<sup>214</sup> However, it can be assumed, based upon the facts of *Whelan*, that much of the program structure's expression was used because of either efficiency concerns or external constraints. Still, a significant amount of protectable expression would probably have remained, because the program's structure was so

---

209. For a discussion of the policies underlying the copyright laws, see *supra* notes 73-83 and accompanying text.

210. For a discussion of the information that is excluded by the Abstraction-Filtration test, see *supra* notes 176-208 and accompanying text.

211. *Computer Assocs.*, 982 F.2d at 710. The Second Circuit noted that after filtering out unprotectable material, in terms of a work's copyright value, what is left is a golden nugget. *Id.*

212. *Id.* at 711 (noting that Abstraction-Filtration test advances policies underlying Copyright Act).

213. For a discussion of the abstraction test, see *supra* notes 44-47 and accompanying text.

214. The *Whelan* court declined to discuss whether parts of the program were constrained by efficiency matters or other external factors. For a discussion of the *Whelan* case, see *supra* notes 111-33 and accompanying text.

large. When printed out, the program structure at issue in *Whelan* comprised over 200 pages of text.<sup>215</sup> Additionally, there were striking similarities throughout the programs' structures. Therefore, while the Third Circuit in *Whelan* may be criticized for its reasoning, the outcome would likely have been the same under the Abstraction-Filtration test.

Similarly, if the *Whelan* test was applied in *Computer Associates*, the Second Circuit likely would have reached the same conclusion. Applying the first part of the *Whelan* test, the ultimate purpose is to create a compatibility component.<sup>216</sup> A compatibility component is quite different from an office organization program.<sup>217</sup> While there may be protectable expression in the choice of what information to include in a program that organizes an office, a compatibility component is designed to interact extensively with an operating system. If there is no other way to express this idea, then the expression merges with the idea and is deemed unprotectable.<sup>218</sup>

### C. Equity Factor Underlying the Abstraction-Filtration Test

By examining the situations in both *Whelan* and *Computer Associates*, the common theme of simple equity arises. The courts' reasoning seems to be result oriented, with both courts using their discretionary powers to achieve equitable results. In *Computer Associates*, Altai had copied verbatim, portions of CA-ADAPTER into OSCAR 3.4.<sup>219</sup> For this action, they were ordered to pay \$364,000 in damages to Computer Associates.<sup>220</sup> At the same time, before this copying occurred, Altai was the owner of a legitimate competitive program.<sup>221</sup> Once the copying was brought to their attention, they attempted to salvage a legitimate program by rewriting the program.<sup>222</sup> Altai incurred all of the costs of developing the new software, and took every possible measure to ensure that CA-ADAPTER was not a source of ideas for the rewritten program. While the idea of a component compatible structure was new to Altai, it is clear that Com-

215. CLAPES, *supra* note 3, at 102. For a detailed discussion of the structure at issue in *Whelan*, see CLAPES, *supra* note 3, at 102-09.

216. For a discussion of the *Whelan* test, see *supra* notes 111-33 and accompanying text.

217. For a discussion of the functions performed by a compatibility component, see *supra* notes 138-42 and accompanying text.

218. *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 715 (2d Cir. 1992) (noting district court's finding that Computer Associates had failed to meet its burden of proof with respect to substantial similarity of parameter lists, macros and other elements of program's structure).

219. *Id.* at 700 (noting that approximately 30% of ADAPTER copied into OSCAR 3.4).

220. *Id.* at 701 (noting that district court awarded \$364,444 in actual damages and apportioned profits).

221. *Id.* at 699. Altai marketed ZEKE without the infringing element, OSCAR 3.4, between 1982 and 1985. *Id.*

222. *Id.* For a discussion of the measures taken by Altai, see *supra* notes 153-59 and accompanying text.

puter Associates could not claim the idea as their own.<sup>223</sup> Therefore, Altai was within its rights to rewrite a program utilizing the idea of component compatibility.

The *Whelan* decision is frequently criticized for defining a program's functions too broadly.<sup>224</sup> However, it is also accepted that the acts of the defendant were blatantly unethical.<sup>225</sup> Therefore, even if they had narrowed the scope of the expression, it is likely that the *Whelan* court would have found infringement.

Although the *Whelan* and *Computer Associates* decisions indicate that courts have a tendency to reach an equitable result by squeezing the particular facts into a legal framework, these two cases alone are not enough to show a distinct pattern. A pattern can only be determined by evaluating other cases that have been decided since *Computer Associates*.

#### V. SUBSEQUENT CASE LAW ADDRESSING COPYRIGHT PROTECTION FOR NON-LITERAL ELEMENTS OF A COMPUTER PROGRAM

Since the Second Circuit developed the Abstraction-Filtration test in *Computer Associates*, three decisions have been based on non-literal copyright protection for computer programs.<sup>226</sup> In each case, the court either explicitly adopted the Abstraction-Filtration test or implicitly adopted its substance. While the analysis involved in the Abstraction-Filtration test provided these courts with a formalistic structure, its application has led to diverse results. Therefore the question remains: What is the underlying determination being made by these courts?

*Gates Rubber Co. v. Bando American, Inc.*<sup>227</sup> was the first case to be decided subsequent to *Computer Associates*. In *Gates*, the United States District Court for the District of Colorado found substantial or nearly identical similarity between all the non-literal elements of the programs at issue.<sup>228</sup> The program at issue made various calculations regarding the configuration of industrial belts as well as the belts' life-expectancy.<sup>229</sup>

223. *Id.* For a discussion of limitations on copyright protection, see *supra* notes 38-47 and accompanying text. Additionally, for a discussion of how the Second Circuit applied these limitations to software, see *supra* notes 180-92 and accompanying text.

224. For a discussion of the criticism of the *Whelan* decision, see *supra* note 133 and accompanying text.

225. Kretschmer, *supra* note 133, at 837 (disagreeing with test used by *Whelan*, and noting that "[o]ne has the impression that the *Whelan* court was eager to find for the plaintiff because of the clearly unscrupulous conduct of the defendant in appropriating the innovative aspects of the Dentalab program").

226. For a discussion of these cases, see *infra* notes 227-40 and accompanying text.

227. 798 F. Supp. 1499 (D. Colo. 1992).

228. *Id.* at 1516. The non-literal elements listed by the district court were: menus, formulas, engineering constants, data flow, control flow, programming style, overall structure and organization of individual modules, and organization of data. *Id.*

229. *Id.* at 1502-03.

The defendant was previously employed by Gates, but claimed that he did not take a copy of the program when he left.<sup>230</sup> The district court discussed the limitations of merger and *scenes à faire*, but failed to eliminate any of the non-literal elements.<sup>231</sup> The extensive non-literal copying, combined with the court's determination that the defendant's testimony lacked credibility, seemed to be sufficient evidence to uphold infringement, regardless of whether the material was protectable.<sup>232</sup>

After *Gates*, the United States District Court for the District of Massachusetts decided *Lotus Development Corp. v. Borland International, Inc.*<sup>233</sup> The court defined the programs idea very broadly, which would have seemed to foreclose any copyright protection.<sup>234</sup> While Borland International (Borland) was free to copy Lotus Development's (Lotus) ideas, the district court recognized that Borland was trying to pilfer the time and effort that Lotus expended in creating a workable interface between the application program and the user. Borland's pirating efforts were accomplished by copying the particular menu hierarchy used, as well as the words to represent the functions within the hierarchy.<sup>235</sup> The district court reasoned that while having a standard interface throughout the industry was a legitimate goal, misappropriating a copyrighted standard was not the appropriate method to achieve this goal.<sup>236</sup> However, by adopting a broad definition as to the idea of Lotus' program, the district court was forced to base its finding of substantial similarity on the actual word choices within the menu hierarchy. As such, the protection afforded Lotus was fairly transparent, even though Borland was penalized for stealing Lotus' hierarchy.

230. *Id.* at 1504.

231. *Id.* at 1519-20.

232. *Id.* at 1520 (noting that during cross-examination defendant "placed a large shadow" over his version of the facts and stating that court's decision was partially "[b]ased on the lack of credibility of [d]efendant").

233. 799 F. Supp. 203 (D. Mass. 1992). The district court adopted the fundamental principle of the Second Circuit in *Computer Assocs. Id.* at 217.

234. *Id.* at 216. The court defined the idea as:

[A] user interface [that] involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the specific spreadsheet operations available in Lotus 1-2-3 are accessible through the paths of the menu command hierarchy.

*Id.* at 215.

235. *Id.* at 220. The court noted that "the menu commands and the menu command hierarchy look the same in both programs. . . . One enters the same keystroke sequence to perform the same spreadsheet operations in both 1-2-3 and Quattro Pro's emulation mode. They feel the same." *Id.*

236. *Id.* at 214 ("Borland did not obtain the right to expressive aspects of Lotus' command hierarchy merely because . . . the 1-2-3 program revolutionized the spreadsheet market.").

The third case subsequent to *Computer Associates* was the decision by the United States District Court for the Northern District of California in *Apple Computer, Inc. v. Microsoft Corp.*<sup>237</sup> Under the guise of protectable screen-display expression, Apple attempted to monopolize an improvement on an idea that it had already licensed to Microsoft.<sup>238</sup> Microsoft had developed software to compete with Apple.<sup>239</sup> The two programs used "windows" that allowed for intuitive and easy access to various computer functions, such as opening and organizing files.<sup>240</sup> Because the improvement was the only practical way to competitively utilize the license, the district court denied protection.<sup>241</sup>

## VI. CONCLUSION

The Abstraction-Filtration test will protect software adequately because its flexibility allows courts to consider the various technological, economic and equitable factors affecting each particular case. The test may not be the best method for balancing the economic and technological interests involved, but, until Congress changes the laws, the test will suffice.

The advantage of the Abstraction-Filtration test is its ability to adapt to various situations and technological advances. Ironically, this flexibility is also the test's weakness. While the test allows courts to reach equitable results, it does not give software developers a firm rule on which they can rely. Therefore, software developers will be forced to rely on their consciences when deciding what elements of another's work they can copy. As such, developers may be forced to overprotect themselves either

---

237. 779 F. Supp. 1006 (N.D. Cal. 1992).

238. *Id.* at 1015. The allegedly infringing software is Windows 2.03. Previously, Apple sued Microsoft for infringement from an earlier version, Windows 1.0. *Id.* To settle the dispute, the parties entered into an agreement whereby Microsoft was granted a "non-exclusive license of the audiovisual displays in Windows 1.0." *Id.*

The district court also seemed annoyed by Apple's procedural approach to the case. *Id.* at 1016. Apple upset the "best laid plans" by "refus[ing] to join the issues raised in defendants' motions," and by "[s]ticking stubbornly to the 'look and feel' . . . theory of this lawsuit." *Id.* Additionally, the court did not seem pleased when Apple refused to respond to a detailed motion by the defendants, stating that "we do not attempt here to chase every rabbit loosed by defendants' continuing focus on irrelevant detail." *Id.* (citations omitted). This type of aggravation does not bode well for a party if the court is utilizing equity concepts.

239. *Id.* at 1018-19. Microsoft's program is designed to operate on IBM personal computers, whereas Apple's program was built into its own line of computers. *Id.* at 1019.

240. *Id.* at 1018. "Windows" are graphical user interfaces that "allow[ ] the user to see, point to and manipulate graphical images, symbols or words to instruct and interact with the computer to perform . . . functions." *Id.*

241. *Id.* at 1020-46. Throughout its discussion, the *Apple Computer* court utilized an approach similar to the Abstraction-Filtration test, by excluding protection for ideas, merged ideas and *scènes à faire*. *Id.*



through unnecessary licensing agreements or to write elements of a program that can be copied.

*Stephen H. Eland*